

(12) LEVEL II



RADC-TR-81-152
Final Technical Report
June 1981

OPTICAL PROCESSING FOR ADAPTIVE PHASED ARRAY RADAR

Carnegie-Mellon University

David Casasent

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

DTIC
ELECTE
S OCT 6 1981 D
B

ROME AIR DEVELOPMENT CENTER

Air Force Systems Command
Griffiss Air Force Base, New York 13441

81 10 5 179

AD A105124

OPTICAL PROCESSING FOR ADAPTIVE PHASED ARRAY RADAR

DTIC FILE COPY

This report has been reviewed by the RADC Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RADC-TR-81-152 has been reviewed and is approved for publication.

APPROVED:



VINCENT C. VANNICOLA
Project Engineer

APPROVED:



FRANK J. REHM
Technical Director
Surveillance Division

FOR THE COMMANDER:



JOHN P. HUSS
Acting Chief, Plans Office

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC (OCTS) Griffiss AFB NY 13441. This will assist us in maintaining a current mailing list.

Do not return this copy. Retain or destroy.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER RADC-TR-81-152	2. GOVT ACCESSION NO. AD-A105 124	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) OPTICAL PROCESSING FOR ADAPTIVE PHASED ARRAY RADAR.		5. TYPE OF REPORT & PERIOD COVERED Final Technical Report, 1 Jan 80 — 31 Dec 80
7. AUTHOR(s) David/Casasent		6. PERFORMING ORG. REPORT NUMBER N/A
9. PERFORMING ORGANIZATION NAME AND ADDRESS Carnegie-Mellon University Department of Electrical Engineering Pittsburgh PA 15213		8. CONTRACT OR GRANT NUMBER(s) F30602-80-C-0048
11. CONTROLLING OFFICE NAME AND ADDRESS Rome Air Development Center (RADC/OCTS) Griffiss AFB NY 13441		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61102F 2304J405
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Same		12. REPORT DATE June 1981
		13. NUMBER OF PAGES 194
		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) Same		
18. SUPPLEMENTARY NOTES RADC Project Engineer: Vincent C. Vannicola (RADC/OCTS)		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Acousto-optics, adaptive phased array radar, adjunct antenna, correlator, iterative optical processor, optical correlation, optical data processing, optical signal processing, signal processing, time integrating correlator, wavelength diversity processor.		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) Three new optical signal processing techniques for adaptive phased array radar (APAR) signal processing are described. The coherent optical cor- relator uses acousto-optic transducers and computes the far-field noise distribution of the antenna. Digital post-processing is then used to compute the adaptive weights from this information. A new hybrid time and space integrating optical signal processing architecture is advanced and a new adjunct antenna concept is introduced for this processor.		

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

(Cont'd)

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

Item 20 (Cont'd)

Initial acousto-optic experimental results and initial simulations are advanced. An electronic support system and the necessary post-processing issues are described.

The second technique uses an input LED array and fiber optic interconnections with a linear photo detector to realize a vector-matrix processor. With the addition of an electronic feedback system, an iterative optical processor results. This system computes the set of adaptive weights given the covariance matrix of the noise field and the desired steering vector. The design and performance of the system fabricated and its use in an APAR signal processing are described.

A new wavelength diversity processor concept for the iterative optical processor is described in many new algorithms and potential applications of the system are provided.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

TABLE OF CONTENTS

	Page
CHAPTER 1 BACKGROUND	1
1.1 INTRODUCTION	1
1.2 APAR OVERVIEW AND REVIEW	2
1.3 COHERENT OPTICAL CORRELATOR SYSTEM	6
1.4 ITERATIVE OPTICAL PROCESSOR (IOP) CONCEPT	15
1.5 WAVELENGTH DIVERSITY PROCESSOR (WDP) CONCEPT	15
CHAPTER 2 COHERENT OPTICAL CORRELATOR (COC)	18
2.1 INTRODUCTION	18
2.2 TIME INTEGRATING (TI) CORRELATOR CONCEPT	19
2.3 ACOUSTO-OPTIC (AO) CELLS	22
2.4 ADJUNCT ANTENNA CONCEPT.	33
2.5 COC SIMULATOR.	35
2.6 HYBRID TIME AND SPACE INTEGRATING (TSI) APAR PROCESSOR	47
2.7 ELECTRONIC SUPPORT SYSTEM.	54
2.8 POST PROCESSING.	60
2.9 TI COC EXPERIMENTS	68
2.10 TSI EXPERIMENTS.	72
2.11 SUMMARY AND CONCLUSION	78
CHAPTER 3 ITERATIVE OPTICAL PROCESSOR (IOP).	80
3.1 INTRODUCTION	80
3.2 SYSTEM DESCRIPTION	80
3.2.1 Error Sources	80
3.2.2 General Purpose Processor Considerations.	83

3.2.3	Notation.	84
3.2.4	Complex-Valued Data Handling.	84
3.2.5	Micro-Processor Feedback System	88
3.3	SYSTEM FABRICATION	91
3.3.1	Optical Vector-Matrix Multiplier.	91
3.3.2	Electronic Feedback System.	93
3.3.3	System Operation.	97
3.4	IOP SIMULATOR.	103
3.5	SYSTEM OPERATION	114
3.5.1	System Calibration.	114
3.5.2	IOP Convergence	117
3.6	EXPERIMENTS.	127
3.6.1	Noise Scenario.	128
3.6.2	M And Mask Determination.	128
3.6.3	Experimental Data	132
3.6.4	Analysis.	133
3.7	SUMMARY AND CONCLUSION	136
CHAPTER 4	WAVELENGTH DIVERSITY PROCESSOR	137
4.1	INTRODUCTION	137
4.2	GENERAL PROCESSOR DESCRIPTION.	137
4.3	EXPERIMENTAL WAVELENGTH MULTIPLEXED PROCESSOR.	143
4.4	REVIEW OF COMPLEX AND BIPOLAR DATA HANDLING.	147
4.5	COMPLEX-VALUED WAVELENGTH-DIVERSITY PROCESSOR.	153
4.6	MULTICHANNEL 1-D CONVOLVER	163
4.7	MATRIX-MATRIX MULTIPLICATION	165
4.8	MATRIX-INVERSION	170

4.9 COVARIANCE MATRIX COMPUTATION	171
4.10 SUMMARY	173
CHAPTER 5 SUMMARY AND CONCLUSION.	175
REFERENCES.	179

Accession For		
NTIS GRA&I	<input checked="checked" type="checkbox"/>	
DTIC TAB	<input type="checkbox"/>	
Unannounced	<input type="checkbox"/>	
Justification		
By		
Distribution/		
Availability Codes		
Avail and/or		
Dist	Special	
A		

LIST OF FIGURES

		Page
Figure 1.1	Simplified block diagram of an adaptive array.	4
Figure 1.2	Diagram of one type of adaptive control loop.	4
Figure 1.3	General form of the output of a multi-channel space integrating correlator for APAR.	10
Figure 1.4a	Isometric view of the output in Fig. 1.3.	11
Figure 1.4b	$\hat{N}(\theta)$ after scanning along the shift in Fig. 1.3.	12
Figure 1.5	Schematic diagram of the iterative optical processor (IOP).	17
Figure 1.6	Schematic diagram of the wavelength diversity processor (WDP).	17
Figure 2.1	Time integrating correlator.	21
Figure 2.2	Multi-channel time integrating correlator.	21
Figure 2.3	Structure of the AO cells.	24
Figure 2.4	Photograph of one of the AO cells.	24
Figure 2.5	TeO_2 response (f_c and bandwidth) versus wavelength (λ) of the input light used [29].	25
Figure 2.6	AO cell frequency response (η versus frequency). (Top) before and (bottom) after impedance matching.	31
Figure 2.7	Adjunct antenna concept.	34
Figure 2.8	Array element time history for an eleven element array with a target of strength 0.7 at 30° .	39
Figure 2.9	Simulated wideband LFM signal.	43
Figure 2.10	Simplified block diagram of a two-channel adjunct antenna TSI processor.	49

Figure 2.11	Block diagram of a time integrating AO TSI adjunct antenna processor.	51
Figure 2.12	Schematic diagram of the hybrid TSI AO correlator.	52
Figure 2.13	Block diagram of the computer driven electronic support system.	55
Figure 2.14	Block diagram of the hardware electronic support system.	56
Figure 2.15	Self-compensating adaptive antenna post-processing (a) $N_m \left(\theta_m \right)$ pattern, (b) null pattern assuming f_0 , and (c) assuming f_m .	64
Figure 2.16	Schematic diagram of a TI AO correlator for residue arithmetic processing for APAR.	70
Figure 2.17	Output plane pattern from Fig. 2.14 for a decimal input $J = 437$ and moduli $m_i = 7, 9$, and 10 .	70
Figure 2.18a	Power spectrum for the first noise source.	73
Figure 2.18b	Power spectrum for the second noise source.	74
Figure 2.19	Wide-band correlation of two noise sources in the TSI system.	75
Figure 2.20	Multi-channel narrow-band correlation of two noise sources in the TSI system.	77
Figure 3.1	Schematic diagram of the iterative optical processor (IOP).	81
Figure 3.2	Signal flow diagram of the bipolar vector-matrix multiplier.	89
Figure 3.3	Schematic diagram of the hardware electronic feedback portion of the IOP.	90

Figure 3.4	Electro-optical vector-matrix multiplier.	92
Figure 3.5	Perspective view of the LED array, fiber optic element and detector.	94
Figure 3.6	Iterative optical processor: hardware block diagram.	95
Figure 3.7	Microprocessor controller.	98
Figure 3.8	Photograph of the entire laboratory IOP system.	99
Figure 3.9	IOP computational cycle flowgraph.	102
Figure 3.10	2-D space-time antenna model (n, n').	105
Figure 3.11	Adaptive antenna pattern for a noise source at 12° (after 100 iterations).	111
Figure 3.12	Adaptive antenna pattern for noise sources at $\pm 30^\circ$ and $\pm 60^\circ$ after 100 iterations.	112
Figure 3.13	Adaptive $E(\theta)$ pattern for noise sources at different angles and velocities.	113
Figure 3.14	Effect of scaling on process gain PG as a function of the number of iterations.	124
Figure 3.15	Process gain (PG) versus number of iterations (k) for different jammer power P_1 for $N = 0.01$ and $P_0 = 0.01$.	125
Figure 3.16	Process gain (PG) obtainable (after 1000 iterations) for different angular separations between the target and jammer for $P_1 = 1$, $P_0 = 0.01$ and $N = 0.01$.	126
Figure 3.17	2-D to 1-D antenna model used for scenario.	129
Figure 3.18	Photograph of the optical mask.	131
Figure 3.19	IOP outputs after iterations 0, 5, 50 (steady state). Each division vertically is 0.5	135

Figure 4.1	Schematic of the wavelength diversity APAR processor.	138
Figure 4.2	Format and notation for P_A and P_B patterns and for the data incident on P_C .	138
Figure 4.3	Functional block diagram of the wavelength diversity APAR processor.	140
Figure 4.4	General patterns for the wavelength diversity APAR processor.	142
Figure 4.5	Schematic of the experimental wavelength diversity APAR processor.	142
Figure 4.6	Schematic of basic vector-matrix non-coherent APAR processor.	147
Figure 4.7	Original color-multiplexed vector-matrix processor schematic.	151
Figure 4.8	Output detector post-processing at P_3 of Fig. 2.7 to realize $\vec{y}_0, \vec{y}_1, \vec{y}_2$ from the $\vec{c}_0, \vec{c}_1, \vec{c}_2$ outputs.	151
Figure 4.9	Formats for the novel complex-valued data processor for APAR.	156
Figure 4.10	Example of complex-valued vector-matrix multiplication.	159
Figure 4.11	Geometrical construction to determine the product of complex vectors as projection components $\hat{0}, \hat{1}, \hat{2}$.	162
Figure 4.12	General P_C pattern description as a multi-channel 1-D wavelength convolver.	164
Figure 4.13	Example of matrix-matrix multiplication.	164
Figure 4.14	Formats for the data planes in Fig. 4.13 for matrix-matrix multiplication.	166

Figure 4.15	Example of new matrix-matrix multiplication algorithm.	168
Figure 4.16	Functional diagram of the matrix-inversion processor.	170

LIST OF TABLES

	Page
TABLE 1.1 PHASED ARRAY RADAR NOTATION USED	8
TABLE 2.1 AO CELL SPECIFICATIONS	30
TABLE 2.2 FORTRAN ROUTINE TO GENERATE CORRELATED AND UNCORRELATED SINE WAVE SIGNALS FOR A TWENTY ELEMENT ANTENNA CORRE- SPONDING TO DIFFERENT ANGULAR NOISE SOURCE LOCATIONS	37
TABLE 2.3 FORTRAN ROUTINE TO GENERATE LFM NOISE	41
TABLE 2.4 ROUTINE TO GENERATE THE RECEIVED SIGNALS FROM GAUSSIAN WHITE NOISE SOURCES	45
TABLE 2.1 IOP NOTATION	85
TABLE 3.2 DATA MEMORY MAP: DM(0 : 63, 0 : 16), 16 BITS	100
TABLE 3.3 PROGRAM MEMORY MAP: PM(0 : 1023), 26 BITS	100
TABLE 3.4 APAR.1. FOR ROUTINE	104
TABLE 3.5 ROUTINE TO COMPUTE S & M	106
TABLE 3.6 ITER.1. FOR ROUTINE	108
TABLE 3.7 ROUTINE TO COMPUTE E(θ)	110
TABLE 3.8 ANTENNA WEIGHTS IN STEADY STATE (k = 50)	134

CHAPTER 1 BACKGROUND

1.1 INTRODUCTION

Adaptive phased array radar (APAR) represents one of the most demanding signal processing problems that merits consideration of the use of advanced signal processing techniques such as optical processing. In this report, we summarize the results of a one year advanced study of the use of optical processing for APAR. The basic concepts and the three major optical signal processing (OSP) approaches used are reviewed in Sects. 1.3-1.5. In Sect. 1.2, we provide an overview and summary of the relevant APAR issues. In Chapters 2-4, each of the new OSP techniques is considered in more detail. Our summary and conclusions are advanced in Chapter 5.

In Sect. 1.2, the salient issues of APAR as they apply to our OSP solutions are reviewed. For a more global description of APAR research, see [1]. In Sect. 1.3, we outline our coherent optical correlator (COC) algorithm and approach. In this system, an optical processor is used to compute the far field noise distribution in angle (or space) and time (or frequency) for a phased array. OSP systems using acousto optic (AO) transducers are used to achieve the necessary system realization in the desired performance. A post processor can then compute the set of adaptive weights to apply to the elements of the antenna to null this far field noise pattern. The details of this system and our recent research on it are included in Chapter 2.

In Sect. 1.4, we outline and highlight our second OSP technique for APAR processing. This system is basically an optical matrix-vector multiplier with an

electronic feedback loop. The matrix-vector multiplier is realized by a linear array of LEDs, fiber optic interconnections, a mask and a linear photo detector array with parallel output. The addition of an electronic feedback system results in an optical system capable of solving a general matrix-vector equation. We refer to this as an iterative optical processor (IOP). For APAR applications, this system is used to compute the set of adaptive weights W given the covariance matrix M in the steering vector S . This system has been fabricated and evaluated. It is described in Chapter 3.

The third and final OSP system for APAR uses multiple input wavelengths as an adjunct to the IOP system of Chapter 3. The resultant wavelength diversity processor (WDP) is highlighted in Sect. 1.5 and described in detail in Chapter 4. It greatly enhances the capability of the IOP system and allows the use of many new algorithms and optical data processing operations for diverse APAR and signal processing applications.

1.2 APAR OVERVIEW AND REVIEW

A simplified block diagram of an adaptive phased array processor is shown in Fig. 1.1. The basic concept is to multiply each of the received signals v_n by an appropriate weight w_n and to then sum these products to produce the output

$$E = \sum_n^N w_n v_n. \quad (1.1)$$

If the same weight is applied to all elements, the beam formed is normal to the array and described by

$$E(\theta) = K \frac{\sin\left(\frac{\pi Nd \sin \theta}{\lambda}\right)}{\sin\left(\frac{\pi d \sin \theta}{\lambda}\right)}, \quad (1.2)$$

where d is the element-element spacing, θ is the steering vector of the array, and N is the number of elements. By varying the phases of the weights w_n , the beam can be steered to different directions. This is achieved by selecting the phases such that a signal incident from the desired scan angle adds in phase across the array.

By adjusting the amplitudes and phases of the w_n , the sidelobe levels can be decreased and the effects of other noise sources in the antenna's field of view can be reduced. In such APAR systems, nulls are placed at angles and frequencies in the antenna pattern corresponding to different noise sources. The adaptive control is achieved by separate adaptive loops on each of the antenna elements as shown in Fig. 1.2. The steering signals s_n^* indicate the direction in which the array is steered and hence the location of the antenna's main beam. The z_n values are the correlation of the received array signals v_n and the output $g = \sum w_n v_n$. When $w_n \odot g$ is large, u_n and hence w_n change rapidly. The purpose of an adaptive array is to reduce the noise in g . When u_n in a given channel is large, the corresponding weight w_n has a larger effect on reducing the residue noise in g .

For receiver noise only, the same for all channels, the weights w_n all approach the same value. If one w_n is larger, the corresponding $w_n v_n$ term in g is larger and this will cause larger z_n and u_n values for that channel, which will decrease w_n . Thus, an adaptive array achieves uniform illumination only for receiver noise alone. When the received signal energy is less than the interference and noise energy, the adaptive loops attempt to minimize the input power (subject to the steering vector constraint). If the noise in g is approximately zero, the z_n values will be small and the w_n will be relatively constant.

The weights can be described by

$$w_n = G \left(s_n^* - u_n \right). \quad (1.3)$$

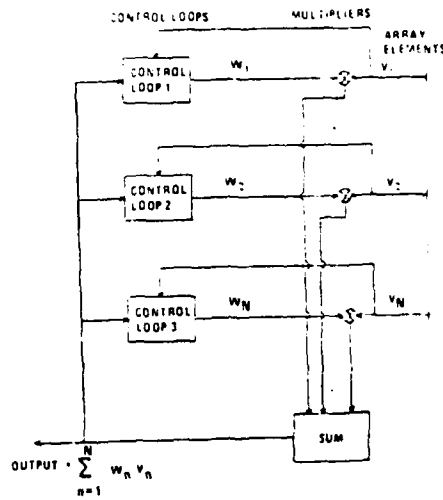


Figure 1.1 Simplified block diagram of an adaptive array.

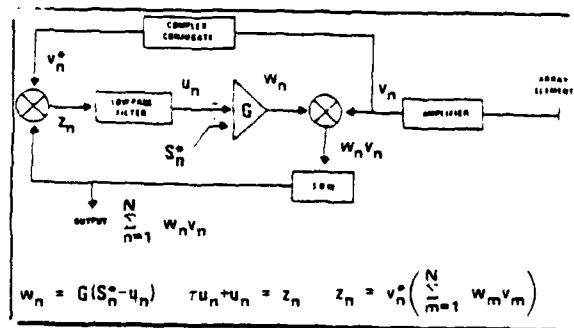


Figure 1.2 Diagram of one type of adaptive control loop.

The low pass filter with time constant τ can be described by

$$\tau \dot{u}_n + u_n = z_n, \quad (1.4)$$

and

$$z_n = v_n^* g = v_n^* \sum w_n v_n. \quad (1.5)$$

These equations describe the response of each of the N adaptive loops. The variation of w_n depends on the inputs from all N channels and on the weights of the other $N-1$ channels. In matrix form, we describe the loops by

$$\begin{aligned} W &= G (S^* - U) \\ \tau \dot{U} + U &= Z \\ Z &= V^* (V - W), \end{aligned} \quad (1.6)$$

where V_T is a row vector and W etc. are column vectors. In terms of average values

$$Z = \overline{V^* V^T} W = MW, \quad (1.7)$$

where M is the covariant matrix

$$M_{ij} = \text{Avg} \left(V_i^* V_j \right). \quad (1.8)$$

If S is a constant steering vector, $\dot{U} = -\dot{W}/G$ and the loop is described by

$$\tau \dot{W}/G + (M + I/G) W = S^*, \quad (1.9)$$

where I is the identity matrix. Equation (1.9) describes a set of N different equations that describe the average response of the array weights W . The values W depend upon the external noise (through M) and the control loop parameters (S^* , G and τ). The conventional approach is to solve (1.9) for W and then apply these W_n to the received array signals.

The covariance matrix M is fundamental to all APAR processing theory since it describes the noise environment. The diagonal terms in M are a measure of the power in each channel, whereas the off-diagonal elements of M describe the direction of arrival of the noise. The rate of convergence of (1.9) depends upon the noise environment. The steady state solution to (1.9) is found with $\dot{W} = 0$ (assuming $G \gg 1$) to be

$$WM = S^*. \quad (1.10)$$

The desired weights W can then be found from S and M^{-1} as

$$W = M^{-1} S^*. \quad (1.11)$$

Equation (1.11) can be obtained by inverting the matrix M , and forming the indicated vector/matrix product. Computation of M requires $2N$ samples at the signal frequency to yield acceptable statistics.

Various correlation loop processors exist. The maximum SNR circuit is the most popular. It uses the Widrow least mean square circuit. It requires an initial estimate for W , computation of the gradient at different surface points, etc. until the minimum concave surface is obtained. The system in Fig. 1.2 uses a typical Howells-Appelbaum loop in which the residue is fed back and correlated with each of the received signals. The filtered outputs are proportional to the gradient. When subtracted from the steering signal, they yield the adaptive weights. A modified random search technique with many variations also exist.

1.3 COHERENT OPTICAL CORRELATOR SYSTEM

In the COC system, we compute the angular and temporal noise distribution of the far field antenna pattern using acousto optic correlators. A dedicated hardware digital post processor can then compute the set of adaptive weights from the

optically computed $N_m(\theta_m)$ noise power N_m of the various noise sources at angles θ_m or $N_m(\theta_m)$, f_m (noise power versus angle and frequency distribution) patterns. The details of this system and our recent research on it are included in Chapter 2. In this section, we highlight the system and algorithm concepts involved for background purposes.

Since the basic concept by which we determine the power N_m and the location angle θ_m of each noise source is quite different from the conventional approach, we briefly review the system philosophy here.

We assume a phased array radar with N receiving elements with parameters given in Table 1.1. The signal s_n received at antenna element n is correlated with a reference signal (the central element $n = N/2$ of the array is used as the reference r for simplicity). We denote the total received signal at element n due only to the noise source m at angle θ_m by f_{nm} . The total received signal at element n due to all M sources is then

$$f_n = \sum_m^M f_{nm}. \quad (1.12)$$

With respect to the reference element r , the signal received at n can be described by

$$f_n(t) = \sum_m^M f_{nm} \left(t - \tau_{nm} \right) \quad (1.13)$$

where

$$\tau_{nm} = k(r-n)d \cos \theta_m \quad (1.14)$$

where $k = 2\pi/\lambda$. For a single noise source m at θ_m , the signal received at elements $N/2 = r$ and n are the same; they are simply delayed in time by τ_{nm} . With M noise sources present, we simply sum over m to obtain the total signal f_n received at array element n .

TABLE 1.1 PHASED ARRAY RADAR NOTATION USED

N	Number Of Phased Array Elements
n	A Given Phased Array Element
f_n	Received Signal At Element n
f_{nm}	f_n Due To A Source At Angle θ_m
d	Center-To-Center Separation Of Array Elements
λ	Wavelength Of Radiation
k	Wave Number, $k = 2\pi/\lambda$
M	Number Of Noise Sources
m	A Given Noise Source
θ_m	Angle Of Noise Source m
s_m	Noise Source m At Angle θ_m

The time dependence of the received signals was explicitly included in (1.13) to allow the delay in arrival time of the signal at two elements of the array to be more conveniently written as τ_{nm} . Of utmost importance is the fact that the delay between elements r and n is also a function of the angle of incidence θ_m or the variable m of the noise source. This delay τ_{nm} is known in advance (for a given array element n and direction θ_m), thus we know where to look for a given channel n and angle θ_m .

The reference signal r sees the sum of M noise sources

$$f_r(t) = f_{r1}(t) + f_{r2}(t) + \dots + f_{rM}(t). \quad (1.15)$$

At array element n , we find

$$f_n(t) = f_{r1}(t - \tau_{n1}) + f_{r2}(t - \tau_{n2}) + \dots + f_{rM}(t - \tau_{nm}). \quad (1.16)$$

Equations (1.15) and (1.16) express (1.13) and (1.14) in detail. The delay τ depends on the element n as well as the angle θ . The separate delays τ_{im} and τ_{km} are related by $\tau_{im} = K\tau_{km}$. For different array elements n , the delay between the signal received at successive array elements (due to a noise source at a fixed θ) equals the linear relationship shown.

The operations required on the reference array element f_r are to correlate it with all N other received signals f_n . This is the operation we propose to optically perform. We denote the correlation of the reference signal f_r and all other N signals by C_{rn} where

$$C_{rn} = C_{r1}(t - \tau_{n1}) + C_{r2}(t - \tau_{n2}) + \dots + C_{rN}(t - \tau_{nN}). \quad (1.17)$$

Each of these correlations $f_r \otimes f_1$, $f_r \otimes f_2$, etc. contains many terms. All cross terms are zero if the separate noise sources are assumed to be independent or not coherent. The result of the correlation of all pairs of correla-

tions for all m thus yield the autocorrelation of all M noise sources with each correlation located at its respective τ value. Since τ is related to θ_m by (1.14), the correlation of f_r and all f_n yields the desired information, the power and angular distribution of all noise sources in the antenna's far field pattern, i.e.

$$N_m \left(\tau_m \right).$$

In Fig. 1.3, we shown the general form for such a multi-channel correlation output. Slit integration (with a specially-shaped detector or by post processing) can provide the desired output information. In Fig. 1.4a, we show the output pattern from such a system in the form of Fig. 1.3. After slit integration, the pattern of Fig. 1.4b results with two peaks at the correct spatial locations corresponding to the angular locations of the two noise sources and with the amplitude of each proportional to the power of the corresponding noise source.

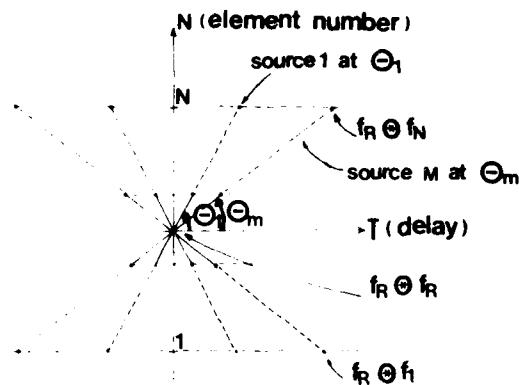


Figure 1.3 General form of the output of a multi-channel space integrating correlator for APAR.

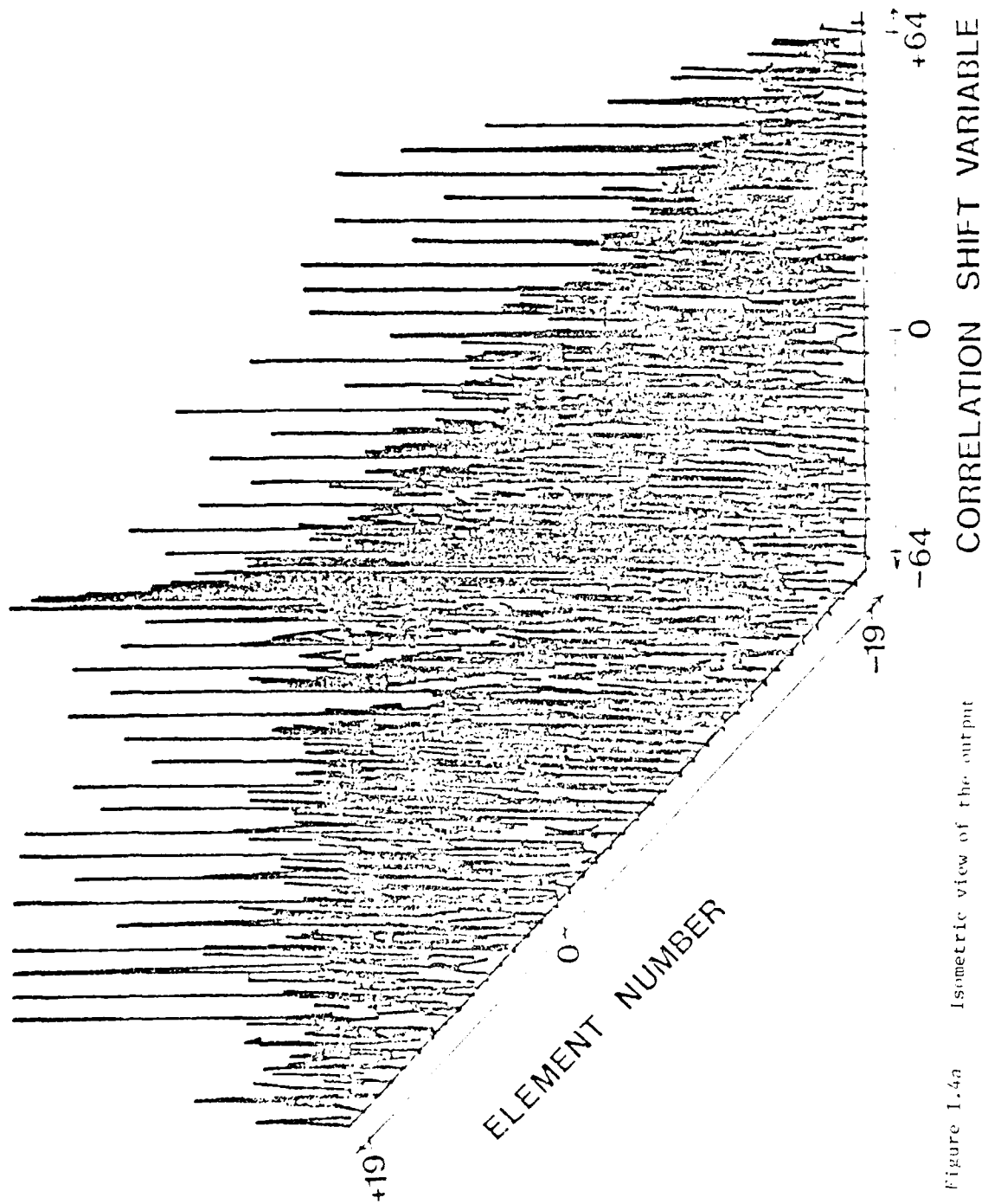


Figure 1.4a Isometric view of the output
in Fig. 1.3.

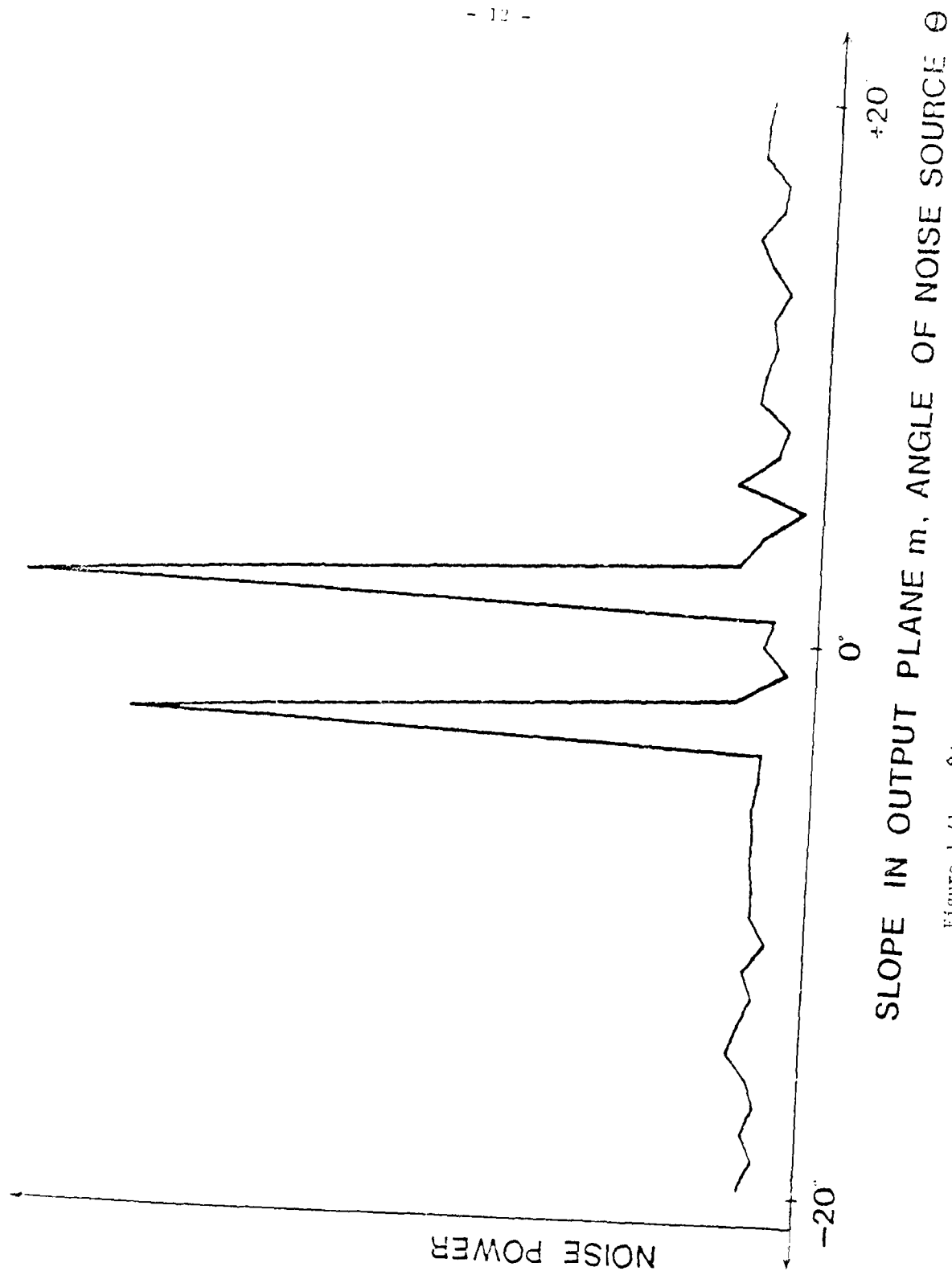


Figure 1.6b $\hat{S}(\epsilon)$ after scanning along the shift in Fig. 1.3.

The details of this approach to APAR processing are described elsewhere [2-4]. For review, we highlight how this correlator approach to APAR processing relates to the more conventional techniques. In Fig. 1.3, the horizontal axis is τ and the vertical axis is array element number n . The horizontal row of spots at $n = N$ in Fig. 1.3 represent the correlation $f_r \circledast f_N$. Since $f_r = \sum_{m=1}^M f_{rm}$ and $f_N = \sum_{m=1}^M f_{Nm}$ are each composed of the sum of the signals received from the M noise sources and since these M noise sources are independent, the location of the M spots along the horizontal line at $n = N$ occur at the τ_{rm} values given by (1.14). These locations are thus proportional to the M angles θ_m of the M noise sources. The intensities of these M spots are proportional to the energies present in each of the M noise sources. These M spots of light are thus the M autocorrelations of the M noise sources, each located at $\tau = \tau_{nm}$ given by (1.14), and thus are proportional to the θ_m noise source angles.

We now show the equivalence of these optical and analytical formulations. Recall that the signals received at array elements 1 to N are f_n and the portion of f_n due to a noise source at θ_m is given by (1.12). For a fixed θ_m , each element n of the array sees the same signal f_m delayed in time by a different amount τ_{nm} at each detector, i.e.,

$$f_{nm}(\tau) = f_m\left(\tau - \tau_{nm}\right) \quad (1.18)$$

where τ is seen to depend on the array element n and the angle θ_m of the noise source. From (1.12) and (1.18), we can write the received signal at element n as in (1.13) as the sum, over the M source angles, of the signals f_{nm} received at array element n .

Element (i, j) of the covariance matrix M is $\overline{f_i f_j}$ or the average over time of the product of the signals received at array elements i and j . We write this as

$$M_{ij} = \overline{f_i f_j} = \int f_i(t) f_j^*(t) dt. \quad (1.19)$$

Substituting (1.13) into (1.19) we find (with t_{jm} the zero reference)

$$M_{ij} = \sum_m \int f_m(t - t_{im}) f_m^*(t) dt. \quad (1.20)$$

Without the t_{jm} zero reference, we have

$$\begin{aligned} M_{ij} &= \sum_m \int f_m(t - t_{im}) f_m^*(t - t_{jm}) dt \\ &= \sum_m \int f_m(t - (t_{im} - t_{jm})) f_m^*(t) dt \\ &= \sum_m C_m(t_{ij}). \end{aligned} \quad (1.21)$$

where $t_{ij} = t_{im} - t_{jm}$. Thus element (i,j) of M is the sum over M of the correlations of the received signals evaluated at t_{ij} .

For a fixed target angle θ_m each element of the covariance matrix M is thus the autocorrelation of the signal (due to that fixed m) at $t_{im} - t_{jm}$. Different elements in M correspond to different array elements and thus different t shifts. For each source m , each row of M is the autocorrelation of that noise source for all t and a given time. Thus $M = M(t, t)$. As shown in (1.21) and noted above, the covariance matrix M is equivalent to a correlation as used in (1.17) and Figs. 1.3-1.4. If we correlate the received signal at one element of the array with the received signals at the other elements of the array we thus obtain (1.21) and thus M . In the optical system, t is continuous of course.

The above analysis has shown that, when written in our correlation terms, the conventional method of APAR processing is equivalent to ours. In Chapter 2, we describe our recent research approaches, algorithms and techniques to realize such an APAR processor in real time with presently available AO transducers.

1.4 ITERATIVE OPTICAL PROCESSOR (IOP) CONCEPT

The basic IOP system is a vector/matrix multiplier (Fig. 1.5) in which the input vector is an estimate of the adaptive weights W_i and is realized as the output from a linear array of LEDs. The matrix $(I - M)$ is a fixed 2-D mask. Associated intermediate optics correctly image W_i onto $(I - M)$ and $I - M$ onto a linear output detector to whose outputs S^* are added. The resultant output

$$W_{i+1} = W_i (I - M) + S^* \quad (1.22)$$

is a new estimate of W . It is fed back to the LED input and the vector/matrix multiplication is repeated until

$$W_i = W_{i+1} \quad (1.23)$$

within some tolerance ϵ . When this occurs, the resultant output

$$W = M^{-1} S \quad (1.24)$$

is the desired solution of (1.23) for W_0 . This system and the associated iterative algorithm used are discussed in more detail in [5,6] and in Chapter 3.

1.5 WAVELENGTH DIVERSITY PROCESSOR (WDP) CONCEPT

An important adjunct to the IOP system is the use of wavelength as an extra input variable dimension. The resultant system is shown schematically in Fig. 1.6 and described in detail in Chapter 4. We refer to this system as a wavelength diversity processor (WDP). As the input to such a system, we use three (or more) linear arrays of laser diodes (LDs). The output from each laser diode uniformly illuminates the corresponding row of the mask as in the IOP. The multi-color light leaving the mask is then split into three or more original input colors by a grating. The corresponding three vector-matrix products in the different wavelengths of light

are thus formed and collected under different corresponding output detectors. Various alternate versions of this system have been developed and demonstrated during the past year. These and the new algorithms and operations possible on such a WDP system are described in Chapter 4.

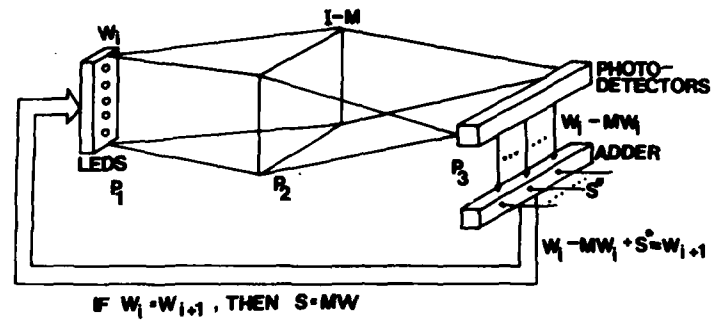


Figure 1.5 Schematic diagram of the iterative optical processor (IOP).

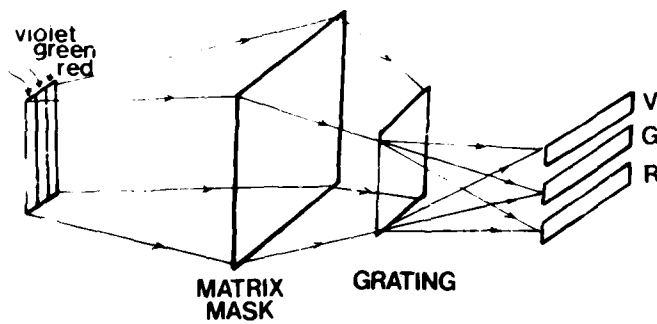


Figure 1.6 Schematic diagram of the wavelength diversity processor (WDP).

CHAPTER 2 COHERENT OPTICAL CORRELATOR (COC)

2.1 INTRODUCTION

In this chapter we describe our recent work on the new coherent optical correlator (COC) concept for APAR signal processing. As noted in Chapter 1, the purpose of this COC system is to obtain an estimate of the far field noise distribution $N_m(\theta_m)$ as a function of the intensity N_m and the angular location θ_m of each source. In our new research, we have altered the original space integrating (SI) correlator system described in Chapter 1 to a time integrating (TI) system (Sect. 2.2). This new system permits longer integration and correlation times and thus better noise field statistical estimates. We have also changed the real time spatial light modulators (SLMs) used from 2-D transducers to 1-D acousto optic (AO) cells. This was done because AO cells are more easily available than 2-D SLMs and because of their higher bandwidth and superior performance, compared to other SLMs. The AO cells we fabricated during this past year are described in Sect. 2.3 together with the initial tests we were able to perform on these devices.

A new adjunct antenna concept for APAR processing on the COC system was formulated (Sect. 2.4) and a digital simulator for this new COC system was written (Sect. 2.5). A new hybrid time and space integrating (TSI) AO system architecture was developed (Sect. 2.6) that extends the COC concept to wideband receivers and wideband noise sources. This new TSI system can compute both the angular (spatial) and temporal (frequency) distribution of the antenna's noise field pattern in parallel. Use of this system can thus enable one to perform adaptivity in both time and space. The two electronic support systems (a computer-driven one and a hardware system)

that we are assembling for the AO COC systems are described in Sect. 2.7. The issue of quantization of the input data and performing complex correlations are key features in the use of both systems. In Sect. 2.8, we present two new post-processing algorithms for the narrow-band and wide-band noise cases. The post-processor is used to compute the adaptive weights from the spatial and temporal field strength distribution of the noise field to be nulled. In Sect. 2.9, experiments on the AO TI correlator for residue arithmetic operations are presented (residue arithmetic was one of the candidate techniques for APAR that we studied in our earlier report [2]). Initial experimental simulations of the hybrid TSI system were then performed (Sect. 2.10) with encouraging results that demonstrated the basic concept. The status of the COC system is then summarized in Sect. 2.11.

2.2 TIME INTEGRATING (TI) CORRELATOR CONCEPT

The prior COC systems we considered were space integrating correlators in which the correlation was performed by multiplication in the frequency domain and the resultant correlation was displayed in space. For these systems, the correlation of the two signals s_1 and s_2 was realized as

$$s_1 \otimes s_2 = \int [S_1 S_2^*] = R_{12}(\hat{x}), \quad (2.1)$$

where capital letters represent the Fourier transforms of the corresponding space functions. In Fig. 2.1, we show the schematic diagram of a time integrating correlator [7]. This is the basic system architecture we propose for the COC processing of adaptive array data. This system realizes the correlation of s_1 and s_2 by integrating in the time domain, i.e.

$$s_1 \otimes s_2 = \int s_1(t) s_2^*(t-\hat{x}) dt = R_{12}(\hat{x}). \quad (2.2)$$

The general description of the simplified system shown is quite direct. An input source (such as a LED or LD) is time sequentially modulated with the received signal s_1 from one antenna element. Thus the output from the LED as a function of time is $s_1(t)$. The output from this point source is expanded to uniformly illuminate an acousto optic (AO) cell fed with an input signal $s_2(t)$. The transmittance of the AO cell is a function of time (t) and spatial location (x) is described by $s_2(t-x/v)$. The light distribution leaving the AO cell is thus $s_1(t) s_2(t-\hat{x})$, where $\hat{x} = x/v$ and where v is the velocity of the acoustic wave in the cell. For simplicity, we set $v = 1$ to simplify the associated mathematics. The AO cell is then imaged onto the output plane where a time integrating detector is placed. This forms the integral of the product of these two signals in time. The resultant integration is the desired correlation in (2.2). It is displayed in space \hat{x} at the output plane. The immediate attractiveness of this optical signal processor for the adaptive radar problem is that long integration times or equivalently long duration signals of large time bandwidth product can be correlated. This system has a low range delay search window (equal to the aperture time, or transit time, τ of the signal across the acousto optic cell). However, this is the exact case that occurs in adaptive radar processing (low range windows and long integration times). Thus this basic correlator architecture appears most appropriate for the adaptive algorithm required.

The system of Fig. 2.1 can easily be extended to a multi-channel time integrating (TI) correlator as shown in Fig. 2.2 by replacing the input light source with a linear array of LEDs or LDs. The resultant output plane pattern now contains N 1-D correlations of the reference signal s_r with all N other received antenna signals. This output pattern is the dual of the one obtained in our original COC system described in Chapter 1. However, no 2-D SLMs are required in this system and it provides longer integration times and hence better noise estimation than the space integrating system.

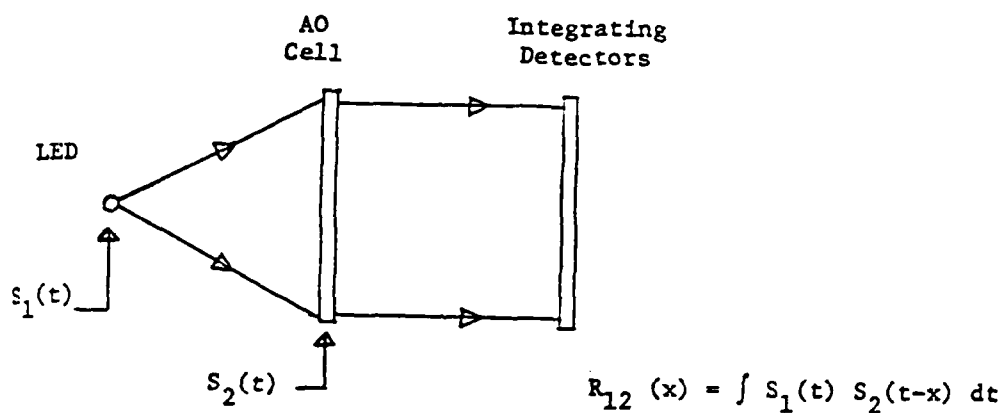


Figure 2.1 Time integrating correlator.

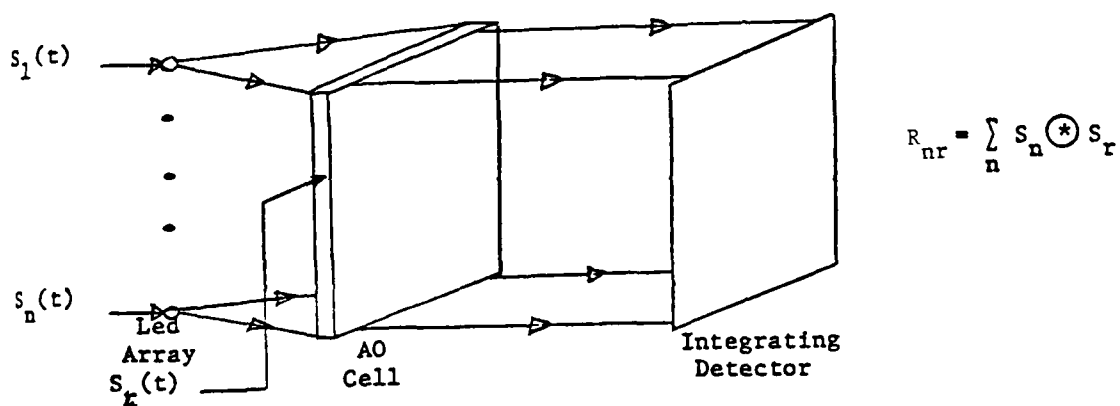


Figure 2.2 Multi-channel time integrating correlator.

2.3 ACOUSTO-OPTIC (AO) CELLS

Considerable time was required to complete fabrication of the AO cells to be used. Much time was also necessary to adequately test the cells. This feature is often ignored in assembly of a system. Since the cell's specifications directly affect the performance of the system and provide necessary data for new improved cell fabrication, such a test program should be completed in the next phase of this program. All AO cell fabrication, polishing and transducer bonding was performed by Westinghouse Corporation (Pittsburgh).

The structure of the AO cells fabricated is shown in Fig. 2.3. Two cells were produced: 30 mm long, 33 mm high and 6 mm thick (see Fig. 2.4). The AO cells consisted of a LNB transducer bonded to the TeO_2 AO material.

Two LNB transducers about $6 \times 14 \text{ mm}^2$ were bonded to the right face and connected in series as shown. An $8.3 \times 10^{-2} \text{ mm}$ thick x-cut LNB crystal is used. The crystal thickness was chosen to yield the desired center frequency f_0 [8] of operation from $t = v_s/2f_0 = 6.6 \times 10^5/2(4 \times 10^7)$. The light wavelength used effects the center frequency and bandwidth of the TeO_2 AO material used as shown in Fig. 2.5. We selected a light source of 633 nm (corresponding to the HeNe laser line) and thus a center frequency of about 40 MHz and a bandwidth of about 20 MHz or more (before matching).

TeO_2 was selected because of its good optical quality, ready availability in large size and its low optical and acoustic attenuation [8]. TeO_2 has a high M_2 figure of merit relating diffraction efficiency η , acoustic power P_a , height of the acoustic beam H and interaction length L by [8]

$$\eta = 5 M_2^2 \lambda_0^2 H^{-1} L P_a. \quad (2.3)$$

For TeO_2 , $M_2 = 795$ compared to $M_2 = 1$ for quartz. Bandwidth is another parameter of concern in AO cells. A second figure of merit $M_1 = M_2 v^2$ describes the band-

width BW performance of the cell, since $BW \propto \eta V^2$. TeO_2 offers an $M_1 = 13.1$ versus $M_1 = 1.0$ for quartz. Other materials such as As_2Se_3 have higher $M_1 = 204$ but are not available commercially. Recent work may change this condition, but more so As_2Se_3 is of use only in the IR region (0.9-11 μm).

A slow shear TeO_2 crystal was thus selected. Its acoustic attenuation in this mode is 16 dB/usec Hz^2 . This is larger than the value obtainable with other materials, but it is much lower (better) than for As_2Se_3 (27.5 dB/usec- Hz^2). For our cell, this parameter is thus negligible since

$$(16) (50 \text{ usec}) (4 \times 10^{-2})^2 \text{ GHz}^2 = 1.28 \text{ dB}$$

or less than a 10% loss in acoustic power across the length of the line. Few other materials (besides KRS-5) allow operation in visible light with good optical quality (KRS-5 is quite soft and thus exhibits poor optical quality). Present technological capability thus dictates the use of TeO_2 .

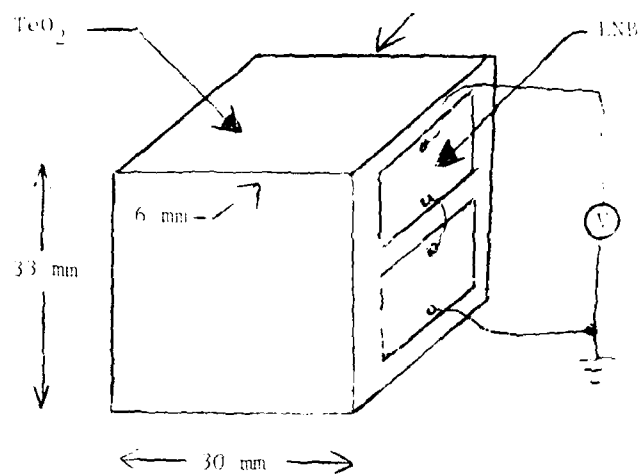


Figure 2.3 Structure of the A0 cells.

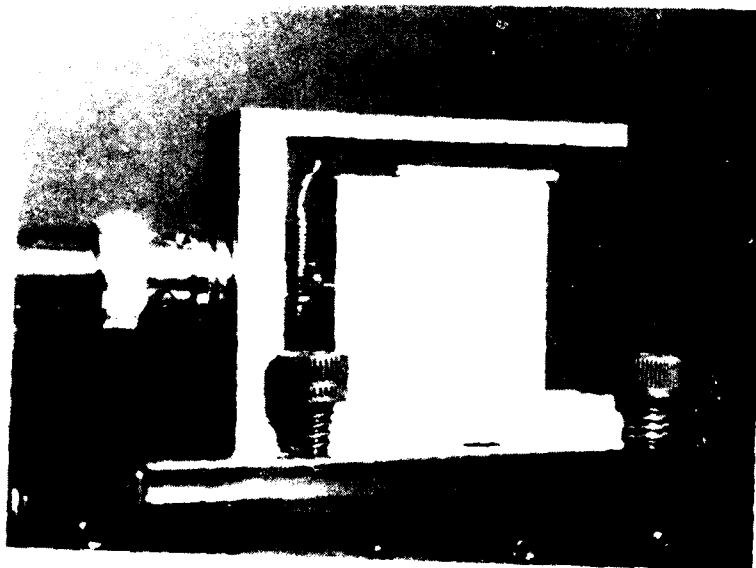


Figure 2.4 Photograph of one of the A0 cells.

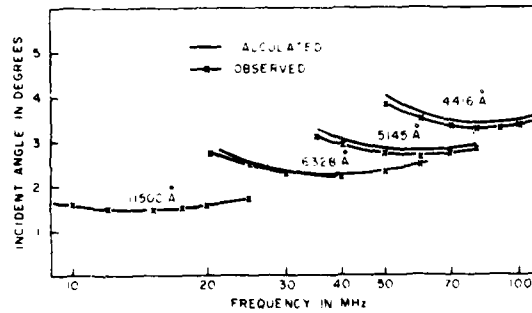


Figure 2.5 TeO_2 response (f_c and bandwidth) versus wavelength (λ) of the input light used [29].

The design and selection of the cells were completed early in month 2. The TeO_2 material, obtained from Crystal Technology, Incorporated, arrived in month 6. The optical faces of the cells were then polished at Westinghouse Corporation. Interferograms were made and polishing was repeated until the two large $30 \times 30 \text{ mm}^2$ faces were flat to $\lambda/2$ over the central 70% of the area. The transducer surface face was then polished to $\lambda/2$. Gold chrome electrodes were deposited on the TeO_2 side of the LNB and the LNB was then cut to the desired size and bonded to the crystal with epoxy and then lapped from its original 0.010" thickness to its final thickness of 3.3 mils.

The epoxy is first mixed to low velocity to allow a 1000 \AA thick layer to be deposited. Small droplets of epoxy are then deposited over the surface (in a clean room) to produce 750 gm/cm^2 deposition and baking at 50°C for 24 hours and cooling at room temperature for eight hours were used. The top Gold electrodes were then deposited and wires were bonded to the top edges of the LNB to obtain the series connection shown in Fig. 2.3. The bottom electrode (Fig. 2.3) is deposited to the TeO_2 before the epoxy bond is applied. These steps were completed in months 7 and 8 at which time the cells were now ready for initial tests.

Initial AO cell tests were performed in month 9. These included initial verification that diffraction occurred, a recheck of the cell's optical quality, and point laser tests of the integrity and uniformity of the LNB bond. These latter frequency response plots are necessary to design the impedance matching network. These tests consisted of illuminating the device quite close to the LNB with a point laser beam. For different positions of the laser beam, the frequency of the input signal was varied and the first order diffracted light intensity versus input frequency was measured. (A single photo diode was used and moved to different positions correspond-

ing to the different diffracted angles for the different input frequencies). After correctly matching cables and cable lengths, the test was repeated with a swept input frequency (20-60 MHz) and the first order diffracted pattern was detected on a reticon linear detector array. The input angle of illumination of the laser light was chosen to yield the flattest spectrum for the transducer-cell combination design.

The first cell had a nice response but with low bandwidth (Fig. 2.6a). The second cell had an anomaly in the response above 50 MHz and a large but non-uniform bandwidth. By varying the laser illumination angle and by proper design of the impedance matching network, we can reduce the central peak response and increase the sidelobe levels. The non-uniform response peaks above 50 MHz do not concern us since we will operate the device at 35-40 MHz with a 10-20 MHz bandwidth.

We now consider the design of the impedance matching network. This absorbed most of our month 10 A0 cell research. The objective was to design a RLC model of the A0 device (transducer and cell) and to include the appropriate RLC impedance matching network to match the frequency response impedance data on the system. A parallel model yielded poor results, with a fairly constant $C = 70-80$ pF value but with a large R range ($\sim 500-50$ ohms) from 25-45 MHz. This large R range would make it difficult to couple input power to the cell and would result in reflected or standing waves. A series model was used and gave better results with: $R = 15.6$ ohms, $C = 71$ pF at 25 MHz; $R = 18.7$, $C = 94$ pF at 40 MHz; and $R = 21$ ohms, $C = 126$ pF at 45 MHz. A smaller R range but an unacceptably large C range occurred. It was thus determined to cancel the capacitive reactance of the system (55 ohms) at 35 MHz with an 0.25 μ H inductor in series with the cell and thus increase the response of the cell at this frequency and at its sidelobes.

In Figs. 2.6a and 2.6b, we show the frequency response of the first AO cell before and after impedance matching. In Fig. 2.6a, we see a resonance peak at $f_c = 47$ MHz, with a quite low 2 MHz bandwidth. In Fig. 2.6b, the impedance matching network is seen to give a much better response with a center frequency of 31 MHz and a 3 dB bandwidth of 18 MHz extending from 22-40 MHz. Similar performance was obtained for the second cell.

In month 11, further initial tests were performed on the device including the measurement of near field point laser beam probing at three vertical locations (center, top and bottom of the 33 mm cell dimension and at 2 mm, 12 mm and 24 mm from the LNB transducer). This data provides necessary AO field strength information from which the usable vertical cell size and the specific vertical portion of the cell to use (from uniformity and/or diffraction efficiency considerations) emerges. The strongest acoustic field (and the largest n by 50%) occurred along the center of the cell. This uneven acoustic field restricts the vertical aperture of use to approximately 1 cm or less in the center of the cell.

In month 12, with the integrity of the optical polish and transducer bond verified, the AO cells were returned to Westinghouse to have an absorption wedge ground onto the left edge of the cell as shown in Fig. 2.4. The angle at the far end of the cell should be chosen to cause reflected waves to diffract back at different angles and in different directions than the original acoustic wave. An angle of 22° degrees was chosen (as a compromise between not decreasing the cell's usable width, not damaging the crystal, and yet producing large angular and directional differences in the reflected waves).

The completed cells arrived just as the present contract period ended and thus only several preliminary final tests were possible. These were conducted after the

final contract date and are included for completeness. The interferogram for cell 1 was repeated and its optical quality was found to be $\lambda/2$ over the central 80% of its area. Scatter level measurements of the bulk properties of the unit showed about a 30 dB scatter level measured at first order. New frequency response data obtained on cell 1 verified an $f_c = 31$ MHz, a 3 dB bandwidth of 18 MHz from 22 MHz to 40 MHz. Unless reflections were very large, we would not expect the f_c and the bandwidth of the cells to change. This was verified by the new data above. A reasonable $\eta = 20\%$ was obtained with 0.5 watt of input electrical power at 31 MHz. Complete tests of η versus f for different input electrical power, input light angle and vertical beam position will be done in a latter phase of this work. Care must be taken not to burn out the cell with large input power. 0.1 watt appears to allow the present cells to perform adequately.

A second AO cell was similarly tested with an 0.25 μ H inductor in series. Its center frequency (27 MHz) and 3 dB bandwidth (14 MHz, from 20 MHz to 34 MHz) were acceptable and compatible with the intended system.

Point probe testing of both cells (or Schlieren imaging of the acoustic wave) were repeated and η versus f was measured as a function of cell position (vertical and horizontal). A 15 mm cell region in the center was chosen for use because it demonstrated the maximum uniformity and maximum η . By measuring the frequency response of the cell for different input powers, the linear operating region of cells 1 and 2 were found to be 0.02 watts to 0.2 watts. We will thus operate cells 1 and 2 over this input power range.

Because of the strong near field effect observed (variation in the intensity of the acoustic interaction versus cell position), no cell phase response measurements of the AO transducer could be made at present. The final operating specifications for the AO cells are summarized in Table 2.1. They appear quite adequate for the intended COC system for APAR.

TABLE 2.1 AO CELL SPECIFICATIONS

<u>Parameter</u>	<u>Cell 1</u>	<u>Cell 2</u>
Time Aperture (Used)	30 μ sec	30 μ sec
Center Frequency	31 MHz	27 MHz
Bandwidth	18 MHz	14 MHz
Time Bandwidth Product	540	420
Optical Quality	$\lambda/2$	$\lambda/2$
Scatter Level	30 dB	30 dB
Diffraction Efficiency	20%	20%
Input Power	0.1 Watt	0.1 Watt

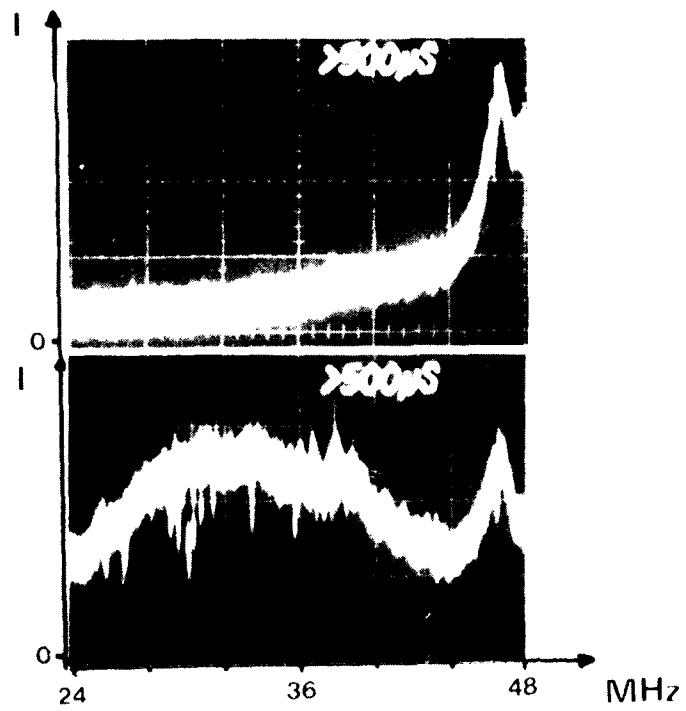


Figure 2.6 AO cell frequency response (I versus frequency). (Top) before and (bottom) after impedance matching.

In the operation of the TI correlator of Fig. 2.1 and its modified forms (Sect. 2.6), care must be taken to match the center frequencies f_1 and f_2 of the input source and the AO cell. In our initial experiments, we will use a point source AO modulator in place of the LED. If $f_1 \neq f_2$, the TI correlator output has a difference term

$$\int s_1(t) s_2(t-\tau) \cos (2\pi\Delta f t) dt$$

and a sum term

$$\int s_1(t) s_2(t-\tau) \cos 2\pi(f_1 + f_2)t dt.$$

If the Δf or $f_1 + f_2$ is greater than the bandwidth of $s_1 s_2$, then the integral approaches zero since integration of an even function for a time much greater than its period is zero. The sum term above will always be zero and we must adjust f_1 to equal f_2 and thus decrease Δf to less than 0.1 of the bandwidth of $s_1 s_2$ or else electronic detection and external integration are necessary. We will thus operate the AO cells slightly off from their central resonance frequencies to ensure $\Delta f = 0$ and that the cosine term in the difference term above approaches 1.

2.4 ADJUNCT ANTENNA CONCEPT

The initial COC concept advanced in [2] and highlighted in Chapter 1 used correlations to obtain the angular location of the target. This operation requires correlation measurements to determine the relative time delays line-to-line between received elements of the phased array. A numerical analysis of the AO cell requirements for such a system were conducted. We assumed a conservative radar center frequency of 1 GHz (corresponding to a $\lambda_R = 0.3$ m) and a linearly spaced phased array with spacing $d = \lambda/2 = 0.15$ m between elements. For a target at an angle $\theta = 60^\circ$, the relative time delay between adjacent elements is thus

$$\tau_d = (d/c) \sin \theta = 0.433 \text{ nsec.} \quad (2.4)$$

If an AO cell with 1 μ sec aperture time (and 1-2 GHz bandwidth) were used, resolving the time delay in (2.4) would require an AO cell TBWP of $1 \mu\text{sec}/0.433 \text{ nsec} = 2300$. Such AO cell specifications [9] are possible but require quite extensive and sophisticated AO cell fabrication and more extensive support electronics (with 1-2 GHz bandwidth). Such efforts were beyond the scope of our present research and the multi-channel system required to process such data is more complex than our simpler 2 channel system.

For these reasons, we chose to consider use of AO cells with larger time apertures (e.g. 40-50 μ sec), lower bandwidths (10-20 MHz), but with comparable TBWP (1000). To use such AO devices for the given APAR problem, we must increase the time delay to be resolved element-to-element in the phased array by the AO correlator. This resulted in the new system concept of an adjunct antenna (with one element widely spaced by many wavelengths from the main phased array). Fig. 2.7 shows the associated adjunct antenna system concept.

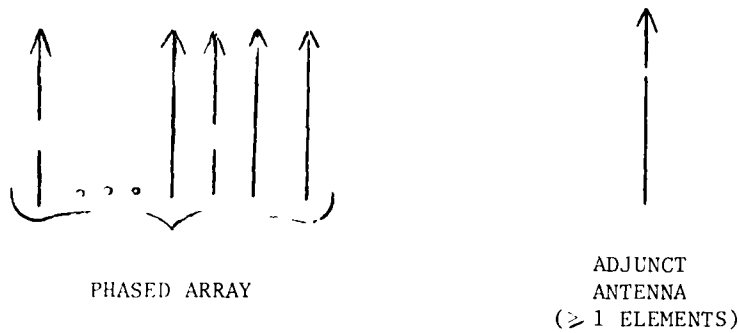


Figure 2.7 Adjunct antenna concept.

Assuming a 20 MHz bandwidth AO cell with a 50 μ sec aperture time (approximately equivalent to the performance of our AO cells) with a TBWP of 1000, the minimum time resolution possible is $50 \mu\text{sec}/1000 = 50 \text{ nsec}$. To realize such a time delay, we must increase the spacing d between adjacent antenna elements (i.e. in this case the spacing between the phased array and the adjunct omni antenna in Fig. 2.7). The time delay τ in (2.4) can be rewritten in units of radians as

$$\tau = (2\pi d/\lambda) \sin \theta. \quad (2.5)$$

The antenna spacing d required to yield a $\tau = 50 \text{ nsec}$ for a target at $\theta = 60^\circ$ ($\sin \theta = 0.866$) is

$$d = \tau c / \sin \theta = 17.3 \text{ m} = 57.6 \lambda \text{ at } f = 1 \text{ GHz}. \quad (2.6)$$

If the input signal is heterodyned to a center frequency $f_c = 40 \text{ MHz}$ or $\lambda = 7.5 \text{ m}$ (corresponding to the center frequency of our AO cell, a normal phased array antenna requires $d = \lambda/2 = 3.75 \text{ m}$ spacing at an associated TBWP of $50,000/10.8 = 4629$. Since this is beyond the scope of presently available AO cells, we consider the use of a adjunct antenna with spacing

$$d = \tau_{\min} c / \sin \theta = (50 \times 10^{-9}) (3 \times 10^8) / 0.866 = 2.3 \lambda = 17.3 \text{ m}. \quad (2.7)$$

This antenna element spacing is quite realistic and is compatible with the TBWP = 1000 obtainable directly with current AO technology.

This adjunct antenna system thus appears capable of resolving the necessary 50 nsec minimum time delays (between the signals received at the phased array and the adjunct antenna element) within the TBWP = 1000 and the 40 MHz heterodyne center frequency of the available AO cells. As we shall see later, processing of the data from such a system requires only a 2 element or 2 channel processor rather than a n channel system (Sect. 2.6).

2.5 COC SIMULATOR

Analysis of the COC processor (Sect. 2.2) and the adjunct antenna concept (Sect. 2.4) were best facilitated by simulation. The signals received at one element of a phased array is the sum of N different noise sources at angles θ_m , range delays τ_m , frequencies f_m and bandwidths B_m . Generation of such composite signals for one or more receiving antenna elements is quite complex considering the multitude of possible noise source parameters.

A COC simulator for APAR was developed with all the necessary features to permit study of advanced phased array systems with space and time diversity. The currently operational simulator can handle 20 adaptive elements (determined by specifying the variable RN). The number of noise sources, their locations and their frequencies are controlled by the simulation parameters RNO, XSOUR, and XFREQ. Similarly, the antenna element spacing is taken to be $\lambda/2$, but can easily be altered within the program (in practice this is done most simply by varying the frequency and hence the wavelength of the input signal). The power levels of the different noise sources are taken to be unity unless otherwise specified. We can also produce random uncorrelated received signals as are necessary in practice by using different phases for different received sinewaves. This can be used to repre-

sent uncertainty in the synchronization of different noise sources as well as producing noise sources with different degrees of coherence.

Let us first consider the case of sine wave signals, since these represent the simplest type of received signal. In this case, the time delay τ between adjacent antenna elements is easily expressed by including a simple phase difference

$$\phi = 2\pi f\tau \quad (2.8)$$

in the received signals. For each frequency, the necessary received signals at different array elements can thus be described by introducing ϕ in (2.8). In our Fortran program, we vary the sampling rate of the input signal to represent different receivers. To verify the fidelity of such monofrequency signals, we take their FFT and observe that it has a non zero value at only one frequency.

To model uncorrelated noise sources, we introduce a small frequency deviation Δf from f such that

$$T\Delta f = \pm n \quad (2.9)$$

where n is an integer and T is the total signal duration. The associated Fortran program is listed in Table 2.2.

In Fig. 2.8, we shown a pseudo isometric display of the received signals at the N elements of the phased array with time and antenna element as independent variables (horizontally and vertically). In this example, 11 independent receiving elements are assumed with a single noise source of amplitude 0.7 at 30^0 with integration over 500 signal points employed. As seen, different sine wave noise sources align along lines at angles corresponding to θ_m of the source. The integration along these lines gives fine peak location accuracy as noted in Chapter 1.

TABLE 2.2

FORTRAN ROUTINE TO GENERATE CORRELATED AND UNCORRELATED SINE WAVE
SIGNALS FOR A TWENTY ELEMENT ANTENNA CORRESPONDING
TO DIFFERENT ANGULAR NOISE SOURCE LOCATIONS

```

00100      !
00200      !
00300      ! THIS PROGRAM GENERATES THE RECEIVED SIN WAVES
00400      ! BY A 20 ELEMENT ANTENNA, TAKING INTO ACCOUNT
00500      ! THE LOCATION OF NOISE SOURCES WITH RESPECT TO
00600      ! ELEMENTS.
00700      !
00800      !
00900      REAL XR(500,20),XI(500,20),XSOUR(20),XFREQ(20)
01000      !
01100      ! PROGRAM INSTRUCTIONS
01200      !
01300      !
01400      2      FORMAT('HOW MANY ELEMENTS (20 MAX.) ?')$
01500      READ(5,100)
01600      8      FORMAT(' ')
01700      WRITE(6,10)
01800      3      FORMAT('HOW MANY NOISE SOURCES (20 MAX) ?')$
01900      READ(5,100)
02000      WRITE(6,10)
02100      4      FORMAT('LOCATIONS IN TERMS OF DEGREES ?')$
02200      READ(5,100)
02300      9      FORMAT(' ')
02400      WRITE(6,10)
02500      6      FORMAT('FREQUENCIES 0.1-1.0 ?')$
02600      READ(5,100)
02700      WRITE(6,10)
02800      !
02900      !
03000      ! VARIABLES
03100      !
03200      !
03300      !
03400      AMPL=0.1
03500      PI=3.141592653589793
03600      SAMPR=100
03700      IQ=0.001
03800      RAAVEL=1.0
03900      ADDEL=
04000      !
04100      !
04200      ! HERE I GENERATE THE REAL AND IMAGINARY PARTS AND
04300      ! STORE THEM IN ARRAYS XR,XI
04400      !
04500      !
04600      INCLUDE 'COMMON'
04700      !
04800      DO 10 I=1,20
04900      XR(I)=0.0
05000      XI(I)=0.0
05100      DO 10 J=1,20
05200      DO 10 K=1,20
05300      DO 10 L=1,20
05400      DELTA=(2*PI*(XSOUR(I)-R-INT)*COSI((XSOUR(JV))*PI)/180))/RAAVEL
05500      DARG=(2*PI*(XSOUR(I)-R-INT)*SINI((XSOUR(JV))*PI)/180))/RAAVEL
05600      XAR=AMPL*(COSI(DELTA)+SINI(DARG))
05700      XAI=AMPL*(SINI(DELTA)+COSI(DARG))
05800      XR(I,IL)=XR(I,IL)+XAR
05900      XI(I,IL)=XI(I,IL)+XAI
06000      20      CONTINUE
06100      10      CONTINUE
06200      111     CONTINUE

```

TABLE 2.2 (Continued)

```
06200      !
06300      !
06400      ! HERE I NORMALIZE SOMEHOW THE MAX.VALUE OF ELEMENTS
06500      ! SO THAT I CAN USE THE PLOT2D
06600      !
06700      !
06800      !
06900      DO 120 I=1,IC
07000      DO 130 IB=1,IA
07100      XR(I,IB)=XR(I,IB)/5
07200      XI(I,IB)=XI(I,IB)/5
07300      130 CONTINUE
07400      120 CONTINUE
07500      !
07600      ! HERE I ADD A BIAS FOR SUBROUTINE
07700      ! PLOT2D
07800      !
07900      DO 60 I=1,IC
08000      DO 70 IMM=1,IA
08100      XR(I,IMM)=XR(I,IMM)+0.3
08200      70 CONTINUE
08300      60 CONTINUE
08400      GO TO 100
08500      DO 30 IBA=1,50
08600      DO 40 IBB=1,IA
08700      WRITE(5,11) IBA,IBB,XR(IBA,IBB),XI(IBA,IBB)
08800      11 FORMAT(1X,13,3X,'ELEMENT ',12,1X,F10.8,5X,F10.8)
08900      40 CONTINUE
09000      30 CONTINUE
09100      199 CALL PLOT2D(500.8,XR)
09200      STOP
09300      END
```

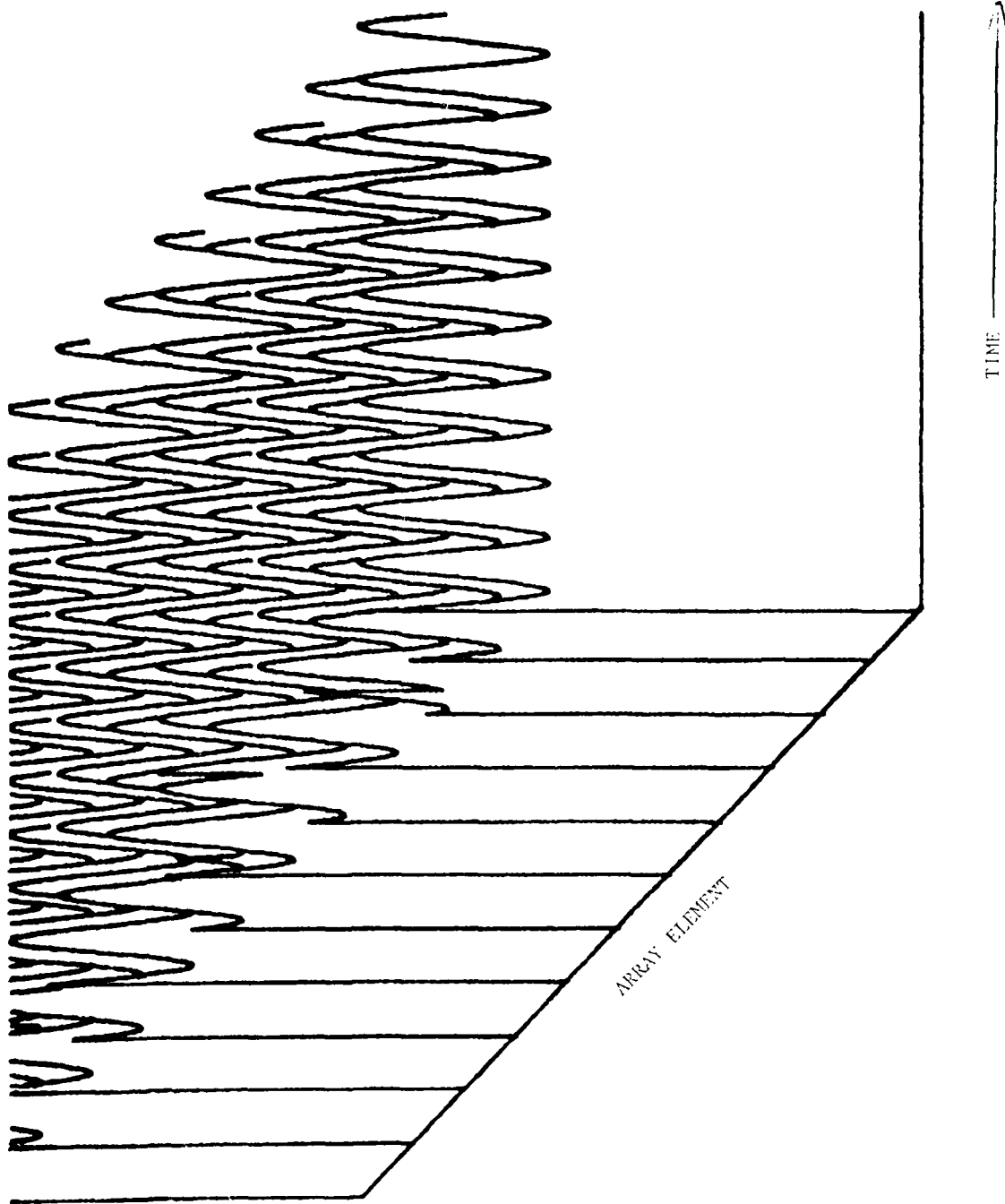


Figure 2.8 Array element time history for an eleven element array with a target of strength 0.7 at 30°.

To describe wide-band noise sources or noise sources with a given bandwidth, our simulator employs LFM signals, since their center frequency and bandwidth can easily be controlled and since their correlation is well predictable. Such signals are most necessary in determining the performance of a APAR system with wide-band signals and wide-band receivers with known deterministic outputs. The Fortran program for this is listed in Table 2.3. It is capable of generating all necessary received signals from multiple sources, each characterized by a different LFM sequence and each with a different center frequency and bandwidth. The initial phase and hence the coherence of the different noise sources can be controlled by randomizing the phase of the different signals. The program parameters XFREQ, XRATE, AMPL and NOSamples determine the parameters of the LFM signal. The LFM waveform produced with XFREQ = 1.0, XRATE = 0.1, AMPL = 1.0 and NO samples = 10.23 is shown in Fig. 2.9 and is seen to be of the desired form.

For such a wide-band signal, time delays between corresponding received elements cannot be represented as simple phase shifts since a given time delay can correspond to different phase delays at different frequencies. Thus, for a given array with element spacing d , d is fixed and in our routine we simulate delayed signals by constructing the corresponding signal and shifting it in the time domain.

To more correctly model true APAR noise sources, random signal sequences with controlled statistical properties rather than the deterministic sine wave and LFM signals considered thusfar are necessary. To describe such noise sources, we specify their amplitude probability density functions and the average signal power (variance) and we characterize their frequency response by their power spectrum. We select the Gaussian PDF as an appropriate choice because it is easily mathematically modelled and because it represents many possible and realistic noise scenarios. Central limit theorem [10] considerations easily verify the appropriateness of such

TABLE 2.3

FORTTRAN ROUTINE TO GENERATE LFM NOISE

```

00100      !
00200      !
00300      ! THIS PROGRAM GENERATES THE RECEIVED LFM WAVES
00400      ! BY A 20 ELEMENT ANTENNA,TAKING INTO ACCOUNT
00500      ! THE NUMBER OF NOISE SOURCES(20 MAX.),THEIR
00600      ! LOCATIONS IN TERMS OF DEGREES,THEIR FREQUENCIES,
00700      ! AND THEIR RATES.
00800      !
00900      !
01000      !
01100      !
01200      REAL XR(1023,20),XI(1023,20),XSOUR(20),XFREQ(20),XRATE(20)
01300      !
01400      ! PROGRAM INSTRUCTIONS
01500      !
01600      WRITE(5,2)
01700      2  FORMAT(1X,'HOW MANY ELEMENTS (20 MAX.) ?')
01800      READ(5,4) RN
01900      8  FORMAT(F)
02000      WRITE(5,3)
02100      3  FORMAT(1X,'HOW MANY NOISE SOURCES (20 MAX.)?')
02200      READ(5,8) RND
02300      WRITE(5,4)
02400      4  FORMAT(1X,'LOCATIONS IN TERMS OF DEGREES ?')
02500      READ(5,9) (XSOUR(I),I=1,20)
02600      9  FORMAT(20F)
02700      WRITE(5,6)
02800      6  FORMAT(1X,'FREQUENCIES 0.1-1.0 ?')
02900      READ(5,9) (XFREQ(I),I=1,20)
03000      WRITE(5,7)
03100      7  FORMAT(1X,'RATES 0.05-0.5 ?')
03200      READ(5,9) (XRATE(I),I=1,20)
03300      !
03400      !
03500      ! VARIABLES
03600      !
03700      !
03800      AMPL=0.1
03900      PI=4.0*ATAN(1.0)
04000      SAMP1=100.0
04100      SAMP2=30.0
04200      IC=10.24
04300      RWAVEL=1.0
04400      AD=0.1
04500      !
04600      !
04700      ! HERE I GENERATE THE REAL AND IMAGINARY PARTS AND
04800      ! STORE THEM IN ARRAYS XR,XI
04900      !
05000      !
05100      DO 10 I=1,RN
05200      10 IF(I) 10,10,10
05300      DO 11 I=1,IC
05400      11 IF(I) 11,11,11
05500      DO 12 I=1,IA
05600      12 IF(I) 12,12,12
05700      DO 13 I=1,IB
05800      13 IF(I) 13,13,13
05900      DO 14 I=1,IC
06000      14 IF(I) 14,14,14
06100      DO 15 I=1,IA
06200      15 IF(I) 15,15,15
06300      DO 16 I=1,IB
06400      16 IF(I) 16,16,16
06500      DO 17 I=1,IC
06600      17 IF(I) 17,17,17
06700      DO 18 I=1,IA
06800      18 IF(I) 18,18,18
06900      DO 19 I=1,IB
07000      19 IF(I) 19,19,19
07100      DO 20 I=1,IC
07200      20 IF(I) 20,20,20
07300      DO 21 I=1,IA
07400      21 IF(I) 21,21,21
07500      DO 22 I=1,IB
07600      22 IF(I) 22,22,22
07700      DO 23 I=1,IC
07800      23 IF(I) 23,23,23
07900      DO 24 I=1,IA
08000      24 IF(I) 24,24,24
08100      DO 25 I=1,IB
08200      25 IF(I) 25,25,25
08300      DO 26 I=1,IC
08400      26 IF(I) 26,26,26
08500      DO 27 I=1,IA
08600      27 IF(I) 27,27,27
08700      DO 28 I=1,IB
08800      28 IF(I) 28,28,28
08900      DO 29 I=1,IC
09000      29 IF(I) 29,29,29
09100      DO 30 I=1,IA
09200      30 IF(I) 30,30,30
09300      DO 31 I=1,IB
09400      31 IF(I) 31,31,31
09500      DO 32 I=1,IC
09600      32 IF(I) 32,32,32
09700      DO 33 I=1,IA
09800      33 IF(I) 33,33,33
09900      DO 34 I=1,IB
10000     34 IF(I) 34,34,34
10100     DO 35 I=1,IC
10200     35 IF(I) 35,35,35
10300     DO 36 I=1,IA
10400     36 IF(I) 36,36,36
10500     DO 37 I=1,IB
10600     37 IF(I) 37,37,37
10700     DO 38 I=1,IC
10800     38 IF(I) 38,38,38
10900     DO 39 I=1,IA
11000     39 IF(I) 39,39,39
11100     DO 40 I=1,IB
11200     40 IF(I) 40,40,40
11300     DO 41 I=1,IC
11400     41 IF(I) 41,41,41
11500     DO 42 I=1,IA
11600     42 IF(I) 42,42,42
11700     DO 43 I=1,IB
11800     43 IF(I) 43,43,43
11900     DO 44 I=1,IC
12000     44 IF(I) 44,44,44
12100     DO 45 I=1,IA
12200     45 IF(I) 45,45,45
12300     DO 46 I=1,IB
12400     46 IF(I) 46,46,46
12500     DO 47 I=1,IC
12600     47 IF(I) 47,47,47
12700     DO 48 I=1,IA
12800     48 IF(I) 48,48,48
12900     DO 49 I=1,IB
13000     49 IF(I) 49,49,49
13100     DO 50 I=1,IC
13200     50 IF(I) 50,50,50
13300     DO 51 I=1,IA
13400     51 IF(I) 51,51,51
13500     DO 52 I=1,IB
13600     52 IF(I) 52,52,52
13700     DO 53 I=1,IC
13800     53 IF(I) 53,53,53
13900     DO 54 I=1,IA
14000     54 IF(I) 54,54,54
14100     DO 55 I=1,IB
14200     55 IF(I) 55,55,55
14300     DO 56 I=1,IC
14400     56 IF(I) 56,56,56
14500     DO 57 I=1,IA
14600     57 IF(I) 57,57,57
14700     DO 58 I=1,IB
14800     58 IF(I) 58,58,58
14900     DO 59 I=1,IC
15000     59 IF(I) 59,59,59
15100     DO 60 I=1,IA
15200     60 IF(I) 60,60,60
15300     DO 61 I=1,IB
15400     61 IF(I) 61,61,61
15500     DO 62 I=1,IC
15600     62 IF(I) 62,62,62
15700     DO 63 I=1,IA
15800     63 IF(I) 63,63,63
15900     DO 64 I=1,IB
16000     64 IF(I) 64,64,64
16100     DO 65 I=1,IC
16200     65 IF(I) 65,65,65
16300     DO 66 I=1,IA
16400     66 IF(I) 66,66,66
16500     DO 67 I=1,IB
16600     67 IF(I) 67,67,67
16700     DO 68 I=1,IC
16800     68 IF(I) 68,68,68
16900     DO 69 I=1,IA
17000     69 IF(I) 69,69,69
17100     DO 70 I=1,IB
17200     70 IF(I) 70,70,70
17300     DO 71 I=1,IC
17400     71 IF(I) 71,71,71
17500     DO 72 I=1,IA
17600     72 IF(I) 72,72,72
17700     DO 73 I=1,IB
17800     73 IF(I) 73,73,73
17900     DO 74 I=1,IC
18000     74 IF(I) 74,74,74
18100     DO 75 I=1,IA
18200     75 IF(I) 75,75,75
18300     DO 76 I=1,IB
18400     76 IF(I) 76,76,76
18500     DO 77 I=1,IC
18600     77 IF(I) 77,77,77
18700     DO 78 I=1,IA
18800     78 IF(I) 78,78,78
18900     DO 79 I=1,IB
19000     79 IF(I) 79,79,79
19100     DO 80 I=1,IC
19200     80 IF(I) 80,80,80
19300     DO 81 I=1,IA
19400     81 IF(I) 81,81,81
19500     DO 82 I=1,IB
19600     82 IF(I) 82,82,82
19700     DO 83 I=1,IC
19800     83 IF(I) 83,83,83
19900     DO 84 I=1,IA
20000     84 IF(I) 84,84,84
20100     DO 85 I=1,IB
20200     85 IF(I) 85,85,85
20300     DO 86 I=1,IC
20400     86 IF(I) 86,86,86
20500     DO 87 I=1,IA
20600     87 IF(I) 87,87,87
20700     DO 88 I=1,IB
20800     88 IF(I) 88,88,88
20900     DO 89 I=1,IC
21000     89 IF(I) 89,89,89
21100     DO 90 I=1,IA
21200     90 IF(I) 90,90,90
21300     DO 91 I=1,IB
21400     91 IF(I) 91,91,91
21500     DO 92 I=1,IC
21600     92 IF(I) 92,92,92
21700     DO 93 I=1,IA
21800     93 IF(I) 93,93,93
21900     DO 94 I=1,IB
22000     94 IF(I) 94,94,94
22100     DO 95 I=1,IC
22200     95 IF(I) 95,95,95
22300     DO 96 I=1,IA
22400     96 IF(I) 96,96,96
22500     DO 97 I=1,IB
22600     97 IF(I) 97,97,97
22700     DO 98 I=1,IC
22800     98 IF(I) 98,98,98
22900     DO 99 I=1,IA
23000     99 IF(I) 99,99,99
23100     DO 100 I=1,IB
23200     100 IF(I) 100,100,100
23300     DO 101 I=1,IC
23400     101 IF(I) 101,101,101
23500     DO 102 I=1,IA
23600     102 IF(I) 102,102,102
23700     DO 103 I=1,IB
23800     103 IF(I) 103,103,103
23900     DO 104 I=1,IC
24000     104 IF(I) 104,104,104
24100     DO 105 I=1,IA
24200     105 IF(I) 105,105,105
24300     DO 106 I=1,IB
24400     106 IF(I) 106,106,106
24500     DO 107 I=1,IC
24600     107 IF(I) 107,107,107
24700     DO 108 I=1,IA
24800     108 IF(I) 108,108,108
24900     DO 109 I=1,IB
25000     109 IF(I) 109,109,109
25100     DO 110 I=1,IC
25200     110 IF(I) 110,110,110
25300     DO 111 I=1,IA
25400     111 IF(I) 111,111,111
25500     DO 112 I=1,IB
25600     112 IF(I) 112,112,112
25700     DO 113 I=1,IC
25800     113 IF(I) 113,113,113
25900     DO 114 I=1,IA
26000     114 IF(I) 114,114,114
26100     DO 115 I=1,IB
26200     115 IF(I) 115,115,115
26300     DO 116 I=1,IC
26400     116 IF(I) 116,116,116
26500     DO 117 I=1,IA
26600     117 IF(I) 117,117,117
26700     DO 118 I=1,IB
26800     118 IF(I) 118,118,118
26900     DO 119 I=1,IC
27000     119 IF(I) 119,119,119
27100     DO 120 I=1,IA
27200     120 IF(I) 120,120,120
27300     DO 121 I=1,IB
27400     121 IF(I) 121,121,121
27500     DO 122 I=1,IC
27600     122 IF(I) 122,122,122
27700     DO 123 I=1,IA
27800     123 IF(I) 123,123,123
27900     DO 124 I=1,IB
28000     124 IF(I) 124,124,124
28100     DO 125 I=1,IC
28200     125 IF(I) 125,125,125
28300     DO 126 I=1,IA
28400     126 IF(I) 126,126,126
28500     DO 127 I=1,IB
28600     127 IF(I) 127,127,127
28700     DO 128 I=1,IC
28800     128 IF(I) 128,128,128
28900     DO 129 I=1,IA
29000     129 IF(I) 129,129,129
29100     DO 130 I=1,IB
29200     130 IF(I) 130,130,130
29300     DO 131 I=1,IC
29400     131 IF(I) 131,131,131
29500     DO 132 I=1,IA
29600     132 IF(I) 132,132,132
29700     DO 133 I=1,IB
29800     133 IF(I) 133,133,133
29900     DO 134 I=1,IC
30000     134 IF(I) 134,134,134
30100     DO 135 I=1,IA
30200     135 IF(I) 135,135,135
30300     DO 136 I=1,IB
30400     136 IF(I) 136,136,136
30500     DO 137 I=1,IC
30600     137 IF(I) 137,137,137
30700     DO 138 I=1,IA
30800     138 IF(I) 138,138,138
30900     DO 139 I=1,IB
31000     139 IF(I) 139,139,139
31100     DO 140 I=1,IC
31200     140 IF(I) 140,140,140
31300     DO 141 I=1,IA
31400     141 IF(I) 141,141,141
31500     DO 142 I=1,IB
31600     142 IF(I) 142,142,142
31700     DO 143 I=1,IC
31800     143 IF(I) 143,143,143
31900     DO 144 I=1,IA
32000     144 IF(I) 144,144,144
32100     DO 145 I=1,IB
32200     145 IF(I) 145,145,145
32300     DO 146 I=1,IC
32400     146 IF(I) 146,146,146
32500     DO 147 I=1,IA
32600     147 IF(I) 147,147,147
32700     DO 148 I=1,IB
32800     148 IF(I) 148,148,148
32900     DO 149 I=1,IC
33000     149 IF(I) 149,149,149
33100     DO 150 I=1,IA
33200     150 IF(I) 150,150,150
33300     DO 151 I=1,IB
33400     151 IF(I) 151,151,151
33500     DO 152 I=1,IC
33600     152 IF(I) 152,152,152
33700     DO 153 I=1,IA
33800     153 IF(I) 153,153,153
33900     DO 154 I=1,IB
34000     154 IF(I) 154,154,154
34100     DO 155 I=1,IC
34200     155 IF(I) 155,155,155
34300     DO 156 I=1,IA
34400     156 IF(I) 156,156,156
34500     DO 157 I=1,IB
34600     157 IF(I) 157,157,157
34700     DO 158 I=1,IC
34800     158 IF(I) 158,158,158
34900     DO 159 I=1,IA
35000     159 IF(I) 159,159,159
35100     DO 160 I=1,IB
35200     160 IF(I) 160,160,160
35300     DO 161 I=1,IC
35400     161 IF(I) 161,161,161
35500     DO 162 I=1,IA
35600     162 IF(I) 162,162,162
35700     DO 163 I=1,IB
35800     163 IF(I) 163,163,163
35900     DO 164 I=1,IC
36000     164 IF(I) 164,164,164
36100     DO 165 I=1,IA
36200     165 IF(I) 165,165,165
36300     DO 166 I=1,IB
36400     166 IF(I) 166,166,166
36500     DO 167 I=1,IC
36600     167 IF(I) 167,167,167
36700     DO 168 I=1,IA
36800     168 IF(I) 168,168,168
36900     DO 169 I=1,IB
37000     169 IF(I) 169,169,169
37100     DO 170 I=1,IC
37200     170 IF(I) 170,170,170
37300     DO 171 I=1,IA
37400     171 IF(I) 171,171,171
37500     DO 172 I=1,IB
37600     172 IF(I) 172,172,172
37700     DO 173 I=1,IC
37800     173 IF(I) 173,173,173
37900     DO 174 I=1,IA
38000     174 IF(I) 174,174,174
38100     DO 175 I=1,IB
38200     175 IF(I) 175,175,175
38300     DO 176 I=1,IC
38400     176 IF(I) 176,176,176
38500     DO 177 I=1,IA
38600     177 IF(I) 177,177,177
38700     DO 178 I=1,IB
38800     178 IF(I) 178,178,178
38900     DO 179 I=1,IC
39000     179 IF(I) 179,179,179
39100     DO 180 I=1,IA
39200     180 IF(I) 180,180,180
39300     DO 181 I=1,IB
39400     181 IF(I) 181,181,181
39500     DO 182 I=1,IC
39600     182 IF(I) 182,182,182
39700     DO 183 I=1,IA
39800     183 IF(I) 183,183,183
39900     DO 184 I=1,IB
40000     184 IF(I) 184,184,184
40100     DO 185 I=1,IC
40200     185 IF(I) 185,185,185
40300     DO 186 I=1,IA
40400     186 IF(I) 186,186,186
40500     DO 187 I=1,IB
40600     187 IF(I) 187,187,187
40700     DO 188 I=1,IC
40800     188 IF(I) 188,188,188
40900     DO 189 I=1,IA
41000     189 IF(I) 189,189,189
41100     DO 190 I=1,IB
41200     190 IF(I) 190,190,190
41300     DO 191 I=1,IC
41400     191 IF(I) 191,191,191
41500     DO 192 I=1,IA
41600     192 IF(I) 192,192,192
41700     DO 193 I=1,IB
41800     193 IF(I) 193,193,193
41900     DO 194 I=1,IC
42000     194 IF(I) 194,194,194
42100     DO 195 I=1,IA
42200     195 IF(I) 195,195,195
42300     DO 196 I=1,IB
42400     196 IF(I) 196,196,196
42500     DO 197 I=1,IC
42600     197 IF(I) 197,197,197
42700     DO 198 I=1,IA
42800     198 IF(I) 198,198,198
42900     DO 199 I=1,IB
43000     199 IF(I) 199,199,199
43100     DO 200 I=1,IC
43200     200 IF(I) 200,200,200
43300     DO 201 I=1,IA
43400     201 IF(I) 201,201,201
43500     DO 202 I=1,IB
43600     202 IF(I) 202,202,202
43700     DO 203 I=1,IC
43800     203 IF(I) 203,203,203
43900     DO 204 I=1,IA
44000     204 IF(I) 204,204,204
44100     DO 205 I=1,IB
44200     205 IF(I) 205,205,205
44300     DO 206 I=1,IC
44400     206 IF(I) 206,206,206
44500     DO 207 I=1,IA
44600     207 IF(I) 207,207,207
44700     DO 208 I=1,IB
44800     208 IF(I) 208,208,208
44900     DO 209 I=1,IC
45000     209 IF(I) 209,209,209
45100     DO 210 I=1,IA
45200     210 IF(I) 210,210,210
45300     DO 211 I=1,IB
45400     211 IF(I) 211,211,211
45500     DO 212 I=1,IC
45600     212 IF(I) 212,212,212
45700     DO 213 I=1,IA
45800     213 IF(I) 213,213,213
45900     DO 214 I=1,IB
46000     214 IF(I) 214,214,214
46100     DO 215 I=1,IC
46200     215 IF(I) 215,215,215
46300     DO 216 I=1,IA
46400     216 IF(I) 216,216,216
46500     DO 217 I=1,IB
46600     217 IF(I) 217,217,217
46700     DO 218 I=1,IC
46800     218 IF(I) 218,218,218
46900     DO 219 I=1,IA
47000     219 IF(I) 219,219,219
47100     DO 220 I=1,IB
47200     220 IF(I) 220,220,220
47300     DO 221 I=1,IC
47400     221 IF(I) 221,221,221
47500     DO 222 I=1,IA
47600     222 IF(I) 222,222,222
47700     DO 223 I=1,IB
47800     223 IF(I) 223,223,223
47900     DO 224 I=1,IC
48000     224 IF(I) 224,224,224
48100     DO 225 I=1,IA
48200     225 IF(I) 225,225,225
48300     DO 226 I=1,IB
48400     226 IF(I) 226,226,226
48500     DO 227 I=1,IC
48600     227 IF(I) 227,227,227
48700     DO 228 I=1,IA
48800     228 IF(I) 228,228,228
48900     DO 229 I=1,IB
49000     229 IF(I) 229,229,229
49100     DO 230 I=1,IC
49200     230 IF(I) 230,230,230
49300     DO 231 I=1,IA
49400     231 IF(I) 231,231,231
49500     DO 232 I=1,IB
49600     232 IF(I) 232,232,232
49700     DO 233 I=1,IC
49800     233 IF(I) 233,233,233
49900     DO 234 I=1,IA
50000     234 IF(I) 234,234,234
50100     DO 235 I=1,IB
50200     235 IF(I) 235,235,235
50300     DO 236 I=1,IC
50400     236 IF(I) 236,236,236
50500     DO 237 I=1,IA
50600     237 IF(I) 237,237,237
50700     DO 238 I=1,IB
50800     238 IF(I) 238,238,238
50900     DO 239 I=1,IC
51000     239 IF(I) 239,239,239
51100     DO 240 I=1,IA
51200     240 IF(I) 240,240,240
51300     DO 241 I=1,IB
51400     241 IF(I) 241,241,241
51500     DO 242 I=1,IC
51600     242 IF(I) 242,242,242
51700     DO 243 I=1,IA
51800     243 IF(I) 243,243,243
51900     DO 244 I=1,IB
52000     244 IF(I) 244,244,244
52100     DO 245 I=1,IC
52200     245 IF(I) 245,245,245
52300     DO 246 I=1,IA
52400     246 IF(I) 246,246,246
52500     DO 247 I=1,IB
52600     247 IF(I) 247,247,247
52700     DO 248 I=1,IC
52800     248 IF(I) 248,248,248
52900     DO 249 I=1,IA
53000     249 IF(I) 249,249,249
53100     DO 250 I=1,IB
53200     250 IF(I) 250,250,250
53300     DO 251 I=1,IC
53400     251 IF(I) 251,251,251
53500     DO 252 I=1,IA
53600     252 IF(I) 252,252,252
53700     DO 253 I=1,IB
53800     253 IF(I) 253,253,253
53900     DO 254 I=1,IC
54000     254 IF(I) 254,254,254
54100     DO 255 I=1,IA
54200     255 IF(I) 255,255,255
54300     DO 256 I=1,IB
54400     256 IF(I) 256,256,256
54500     DO 257 I=1,IC
54600     257 IF(I) 257,257,257
54700     DO 258 I=1,IA
54800     258 IF(I) 258,258,258
54900     DO 259 I=1,IB
55000     259 IF(I) 259,259,259
55100     DO 260 I=1,IC
55200     260 IF(I) 260,260,260
55300     DO 261 I=1,IA
55400     261 IF(I) 261,261,261
55500     DO 262 I=1,IB
55600     262 IF(I) 262,262,262
55700     DO 263 I=1,IC
55800     263 IF(I) 263,263,263
55900     DO 264 I=1,IA
56000     264 IF(I) 264,264,264
56100     DO 265 I=1,IB
56200     265 IF(I) 265,265,265
56300     DO 266 I=1,IC
56400     266 IF(I) 266,266,266
56500     DO 267 I=1,IA
56600     267 IF(I) 267,267,267
56700     DO 268 I=1,IB
56800     268 IF(I) 268,268,268
56900     DO 269 I=1,IC
57000     269 IF(I) 269,269,269
57100     DO 270 I=1,IA
57200     270 IF(I) 270,270,270
57300     DO 271 I=1,IB
57400     271 IF(I) 271,271,271
57500     DO 272 I=1,IC
57600     272 IF(I) 272,272,272
57700     DO 273 I=1,IA
57800     273 IF(I) 273,273,273
57900     DO 274 I=1,IB
58000     274 IF(I) 274,274,274
58100     DO 275 I=1,IC
58200     275 IF(I) 275,275,275
58300     DO 276 I=1,IA
58400     276 IF(I) 276,276,276
58500     DO 277 I=1,IB
58600     277 IF(I) 277,277,277
58700     DO 278 I=1,IC
58800     278 IF(I) 278,278,278
58900     DO 279 I=1,IA
59000     279 IF(I) 279,279,279
59100     DO 280 I=1,IB
59200     280 IF(I) 280,280,280
59300     DO 281 I=1,IC
59400     281 IF(I) 281,281,281
59500     DO 282 I=1,IA
59600     282 IF(I) 282,282,282
59700     DO 283 I=1,IB
59800     283 IF(I) 283,283,283
59900     DO 284 I=1,IC
60000     284 IF(I) 284,284,284
60100     DO 285 I=1,IA
60200     285 IF(I) 285,285,285
60300     DO 286 I=1,IB
60400     286 IF(I) 
```

TABLE 2.3 (CONTINUED)

```

06100      1  +PHASE
06200      XAR=AMPL*COS(DARG)
06300      XAI=AMPL*SIN(DARG)
06400      XR(I,IL)=XR(I,IL)+XAR
06500      XI(I,IL)=XI(I,IL)+XAI
06600      20  CONTINUE
06700      10  CONTINUE
06800      111 CONTINUE
06900      !
07000      ! HERE I NORMALIZE SOMEHOW THE MAX VALUES OF ELEMENTS
07100      ! SO THAT I CAN USE THE PLOT2D
07200      !
07300      DO 120 I=1,IC
07400      DO 130 IB=1,IA
07500      XR(I,IB)=XR(I,IB)/3
07600      XI(I,IB)=XI(I,IB)/3
07700      130 CONTINUE
07800      120 CONTINUE
07900      !
08000      !
08100      !
08200      !
08300      ! HERE I MAKE THE XR ARRAY POSITIVE,BY ADDING A BIAS
08400      ! IN ORDER TO BE ABLE TO USE SUBROUTINE PLOT2D
08500      !
08600      DO 60 I=1,1023
08700      DO 70 IM=1,IA
08800      XR(I,IM)=XR(I,IM)+0.2
08900      70  CONTINUE
09000      60  CONTINUE
09100      GOTO 1094
09200      DO 30 IBA=1,180
09300      DO 40 IBB=1,IA
09400      WRITE(5,11) IBA,IBB,XR(IBA,IBB),XI(IBA,IBB)
09500      11  FORMAT(1X,I3,3X,'ELEMENT ',I2,1X,F10.8,5X,F10.8)
09600      40  CONTINUE
09700      30  CONTINUE
09800      1099 CALL PLOT2D(400.5,XR)
09900      STOP
10000      END

```

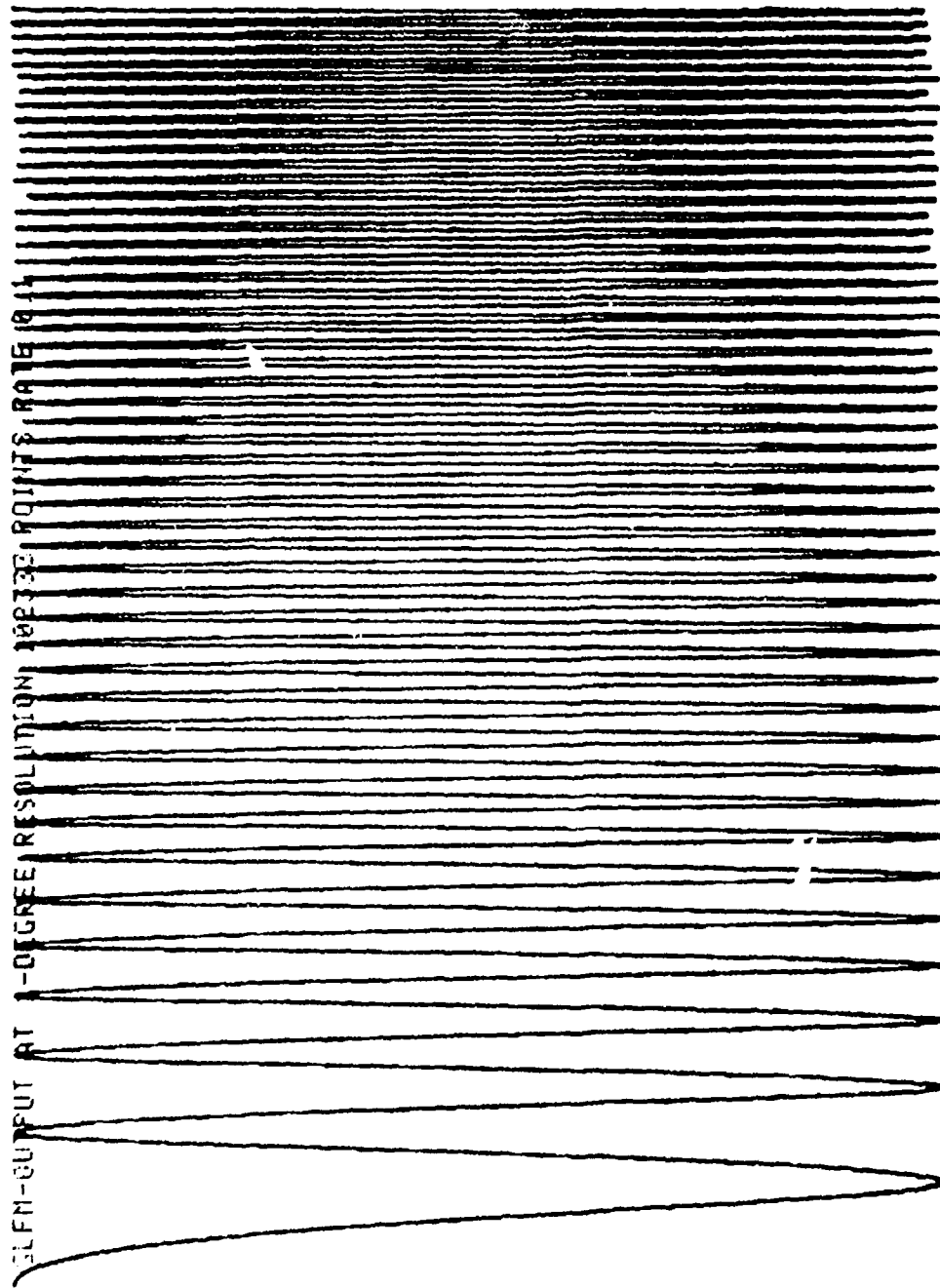


Figure 2.9 Simulated wideband LFM signal.

a model. Various noise source spectra such as white noise can be generated by various digital filters (separate routines exist for this [2]) to produce noise sources of any desired bandwidth.

The Fortran program in Table 2.4 produces the APAR required signals for two Gaussian noise sources. It uses the library function RAN from the DEC operating system. This routine generates random numbers uniformly distributed between 0.0 and 1.0 that are then used to generate a Gaussian random variable sequence as in [11]. The program parameters: RN (number of receiving elements), RNO (number of noise sources), and XSOUR (angular location of the noise sources) are used to control the signal produced. The probability density of the data produced by this routine was verified by generating 10,000 samples of Gaussian noise and inspecting the histogram of the samples to verify its Gaussian shape. With 68% of the samples found to lie between ± 0.40 out of the full ± 1.0 range the data also has the desired standard deviation $\sigma = 0.40$. The fidelity of this random data is also verified by averaging the FFT of many sampled realizations to decrease the estimation variance. A smooth and flat spectrum resulted from this experiment. However, for white random noise, observing that its autocorrelation is a delta function is a simpler test. To vary the bandwidth of such white random noise, digital filters using Butterworth polynomials and bilinear transformations are used as explained in [2].

These new routines, combined with those in [2] now enable us to produce mono, LFM and random signals whose frequency, bandwidth, number, duration, delay and mutual coherence can be independently varied. They now include multiple signals that are adaptive in space and time. These signals typify the type of data to be expected at the receiving elements of a phased array and adjunct antenna and will be the ones used in the COC and TIAO systems both for simulation and in the computer driven electronic hardware support system.

TABLE 2.4

ROUTINE TO GENERATE THE RECEIVED SIGNALS FROM GAUSSIAN WHITE NOISE SOURCES

```

00100      !
00200      !
00300      ! THIS PROGRAM GENERATES THE RECEIVED 'WHITE' NOISE
00400      ! BY A 20 ELEMENT ANTENNA, TAKING INTO ACCOUNT
00500      ! THE LOCATIONS OF NOISE SOURCES (10 MAX) WITH RESPECT TO
00600      ! ELEMENTS.
00700      !
00800      ! NOTE: EACH NOISE SOURCE PRODUCES ITS OWN "WHITE NOISE"
00900      ! ALL THESE WHITE NOISES ARE UNCORRELATED
01000      !
01100      !
01200      REAL XR(1023,20),XI(1023,20),XSDIR(20),XFREQ(20)
01300      !
01400      ! PROGRAM INSTRUCTIONS
01500      !
01600      WRITE(5,2)
01700      2  FORMAT('X, HOW MANY ELEMENTS (20 MAX.) ?')$
01800      READ(5,10) N
01900      8  FORMAT('I')
02000      WRITE(5,3)
02100      4  FORMAT('X, HOW MANY NOISE SOURCES (10 MAX) ?')$
02200      READ(5,10) M
02300      WRITE(5,4)
02400      3  FORMAT('X, LOCATIONS IN TERMS OF DEGREES ?')$
02500      READ(5,10) (XSDIR(I), I=1,10)
02600      9  FORMAT('D')
02700      WRITE(5,5)
02800      6  FORMAT('X, FREQUENCIES 0.1-1.0  ?')$
02900      READ(5,10) (XFREQ(I), I=1,10)
03000      !
03100      !
03200      ! VARIABLES
03300      !
03400      !
03500      PI=3.141592653589793
03600      SATP=48.0
03700      IC=1023
03800      RAA=EL=1.0
03900      AD=0.5
04000      AC=0.15
04100      !
04200      !
04300      ! HERE I GENERATE THE REAL AND IMAGINARY PARTS AND
04400      ! STORE THEM IN ARRAYS XR,XI
04500      !
04600      !
04700      INDIR=1*(RND)
04800      IA=IF(INDIR)
04900      DO 111 I=1,INC
05000      Y1=RA*(0.1,1023)
05100      Y2=RA*(0.1,1023,111111.5)
05200      X1=RA*(0.1,111,111,333)
05300      X2=RA*(0.1,1023,1023,0)
05400      DO 10 J=1,10
05500      Y=RA*(Y1,Y2)
05600      Z=(Y-0.5)*2
05700      PHASE=Z*PI
05800      XR=RA*(X1,X2)
05900      EL=2.7182818
06000      RE=(X+IE**((-(X**2)/(2*(AC**2)))))/(2*(AC**2))

```

TABLE 2.4 (CONTINUED)

```

06100      CO 20 IL=1,IA
06200      R=FLDATT(1)
06300      DEL=(2*PI*AD*(R-RN)+COS(((XSOUR(IM))*PI)/180))/RAVEL
06400      DARG=(2*PI*(XFREQ(IM))*1)/SAMP+PHASE+DEL
06500      XAR=RR+COS(DARG)
06600      XAI=RR+SIN(DARG)
06700      XR(I,IL)=XR(I,IL)+XAR
06800      XI(I,IL)=XI(I,IL)+XAI
06900      20      CONTINUE
07000      10      CONTINUE
07100      111     CONTINUE
07200      !
07300      ! HERE I NORMALIZE SOMEHOW THE MAX VALUES OF
07400      ! ELEMENTS,SO THAT I CAN USE THE PLOT2D
07500      !
07600      DO 120 IB=1,IC
07700      DO 130 IB=1,IA
07800      XR(I,IB)=XR(I,IB)/13
07900      XI(I,IB)=XI(I,IB)/13
08000      130     CONTINUE
08100      120     CONTINUE
08200      !
08300      !
08400      !
08500      ! HERE I ADD A BIAS FOR SUBROUTINE
08600      ! PLOT2D
08700      !
08800      DO 60 IM=1,1023
08900      DO 70 IX=1,IA
09000      XR(I,IM)=XR(I,IM)+0.5
09100      70      CONTINUE
09200      60      CONTINUE
09300      GO TO 1099
09400      DO 30 IBA=1,180
09500      DO 40 IBB=1,IA
09600      WRITE(5,11) IBA,IBB,XR(IBA,IBB),XI(IBA,IBB)
09700      11      FORMAT(1X,13,3X,'ELEMENT ',I2,1X,F10.8,5X,F10.8)
09800      40      CONTINUE
09900      30      CONTINUE
10000      1099   CALL PLOT2D(200,5,XR)
10100      STOP
10200      END

```

2.6 HYBRID TIME AND SPACE INTEGRATING (TSI) APAR PROCESSOR

The basic AO TI correlator was shown in Fig. 2.1 and a multi-channel system in Fig. 2.2. They are described in Sect. 2.2. In this section, we consider a new AO processor that can provide both the angular and temporal noise distribution $N_m(\theta_m, f_m)$ of the antenna's far-field noise pattern. This is necessary when wide-band noise sources and/or wide-band receivers are employed and is necessary to produce a APAR system with adaptivity in both angle and time (i.e. space and frequency).

We concentrate on the adjunct antenna system (Sect. 2.4) for the reasons described earlier, i.e. it allows the necessary target angles to be resolved by correlation. Only a 2 channel processor (e.g. Fig. 2.1) is thus necessary with one signal being the received signal from the adjunct antenna and the second signal being the received signal at the central reference element of the main N element phased array. If a multi-channel system were used (e.g. Fig. 2.2) in which the N phased array received signals were correlated with the reference adjunct antenna element, it would not be possible to resolve the locations of the correlation peaks on the different output channels as noted in Sect. 2.4. Since no added resolvable information is obtained from the different channels, only two are used.

We now consider how to use the TI system of Fig. 2.1 in the adjunct antenna phased array scenario and how to modify it to produce both angle and frequency information on the antenna's far-field noise pattern. The case of a single wide-band noise source simultaneously generating multiple discrete frequencies is considered first because of the notational and conceptual simplicity it provides. The APAR problem is thus effectively reduced to a 2 antenna element case (with a large

d $\lambda/2$ spacing between antenna elements). The two received signals are described by

$$s_a(t) = s(t-\tau/2) \quad (2.10a)$$

$$s_b(t) = s(t+\tau/2), \quad (2.10b)$$

where the time delay (in seconds) between the two received signals is

$$\tau = (d/c) \sin \theta. \quad (2.11)$$

Note that c is the velocity of propagation of the radiation and that τ is independent of the frequency of the radiation and hence the frequency distribution of the noise sources and depends only on the angle θ at which the noise sources are located.

The basic concept used in the TSI processor is to produce frequency filtered versions of the received signal $s(t)$ with different bandpass center frequencies and to correlate these with the original signal $s(t)$. Since the correlation peak value is proportional to the energy of the signal, we can thus determine the amount of signal energy in different frequency bands. [Although similar information can be obtained from a simple Fourier transform of $s(t)$, this will not work when $s(t)$ contains multiple signals at different angles, each with a different bandwidth and frequency distribution]. This concept is shown in block diagram form in Fig. 2.10 where $N + 1$ correlations are produced: N correlations R_1-R_N of the narrow-band filtered signals s_1-s_N (with band pass filters h_1-h_N) with the wide-band reference signal $s(t-\tau/2)$ and the correlation R_T of the wide-band signals s_a and s_b .

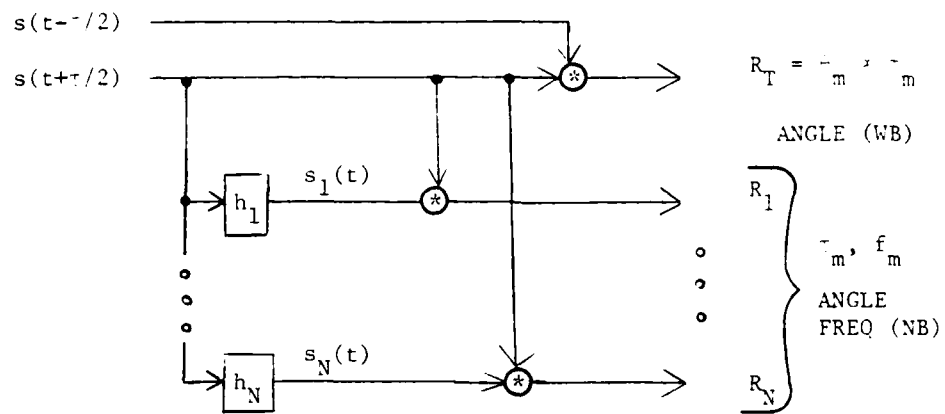


Figure 2.10 Simplified block diagram of a two-channel adjunct antenna TSI processor.

The peak values of R_1 - R_N denote the energy contained in the N frequency bands chosen of the wide-band signal s . If we increase the number of filters N , we increase the frequency resolution. However, increasing N , decreases the width of the h_n filters, thus producing weaker and much wider output correlations, hence making correlation plane detection more difficult. A compromise in N (large N for resolution and small N for accurate detection) is necessary. The choice of N depends on the signal bandwidth, the detector sensitivity, the integration time and the number of adaptive weights available. Since there is no need to estimate the frequency and angular resolution of the far-field noise to a resolution better than that for which the radar can cancel and adapt; the number of adaptive weights available also determine the choice of N .

The output R_T from the top channel is the wide-band correlation of the wide-band signals s_a and s_b . This output will have a much narrower correlation peak width than the other N correlations and is thus used to determine the target's angle θ_m ; whereas the other N correlations provide information on the frequency distribution of the noise at different angles.

To realize the processing depicted in Fig. 2.10, an AO TI correlator is used. The block diagram of such a system is shown in Fig. 2.11. In this system, shifted versions of s_b are produced by a AO cell. Different band pass filtered versions of s_b are produced by frequency plane filtering (BP filts) and correlation is achieved by multiplying (X) the shifted signals and time integrating (TI) the products on an output detector.

A schematic diagram of the AO hybrid TSI correlator is shown in Fig. 2.12. This AO architecture combines the best features of the SI and the TI AO correlators into a new hybrid architecture. This system uses only one real-time 1-D transducer (AO cell), a time sequentially modulated LED light source (or a point AO modulator) and a fixed mask. The received signal $s_a(t)$ is used to time sequentially modulate the output from a LED light source, whose output is expanded to uniformly illuminate an AO cell at P_2 . The AO cell is fed with the signal $s_b(t)$. With s_a and s_b as in (2.10), the light distribution leaving P_2 is

$$u_2(x, y, t) = s(t - \tau/2) s(t + \tau/2 - x) \hat{h}(y), \quad (2.12)$$

where the AO cell produces the continuous shift x and the $\hat{h}(y)$ function notes that this is the pattern at $y = 0$.

Lens L_1 produces the 2-D Fourier transform of (2.12) and incident on P_3 we find

$$u_3\left(\cdot, x, \cdot, y, t\right) = s(t - \tau/2) S\left(\cdot, x\right) \exp \left[j \cdot x (t + \tau/2)\right], \quad (2.13)$$

where the first term is not affected since it is not a function of x and y , where $S\left(\cdot, x\right)$ denotes the Fourier transform of $s(t)$ and where the pattern is independent of y , i.e. the same distribution in (2.13) is present on all vertical lines.

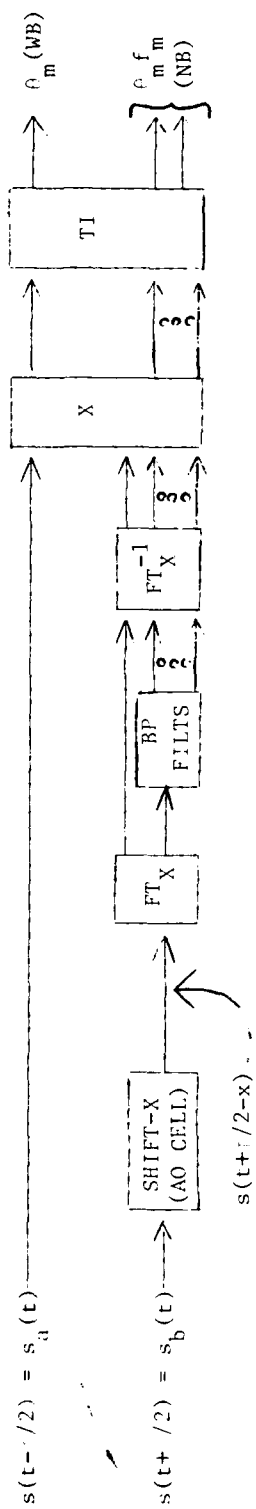


Figure 2.11 Block diagram of a time integrating AO TSI adjunct antenna processor.

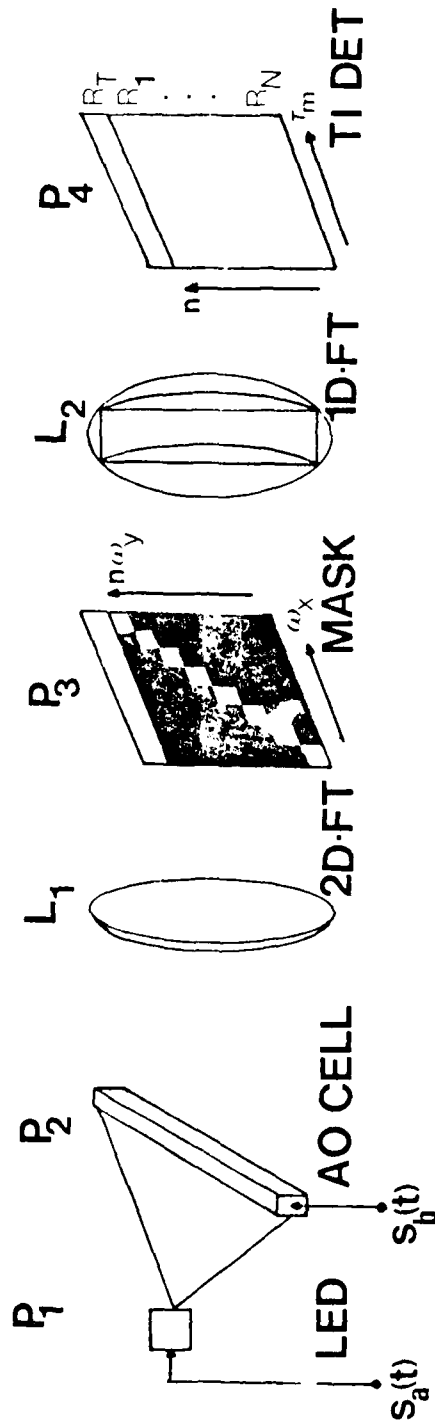


Figure 2.12 Schematic diagram of the hybrid TSI AO correlator.

At P_3 , we see that each vertical line passes a different spatial frequency part of u_3 due to the apertures located at different horizontal positions on each line of the mask. We describe the light leaving the n -th line at P_3 by

$$u_3'(\tau_x, n, t) = u_3 \text{RECT}\left(\tau_x - \tau_n\right), \quad (2.14)$$

where the RECT function describes the band pass filter at P_3 and τ_n is the center frequency of the n -th channel of this band pass filter mask.

Lenses L_2 image P_3 onto P_4 in the vertical direction and produce the Fourier transform of u_3' in the horizontal direction along τ_x . The light distribution incident on P_4 is

$$u_4(\tau_x, n, t) = s(t - \tau/2) s_n(t + \tau/2 - \tau_x), \quad (2.15)$$

where s_n denotes the n -th narrow-band filtered version of $s(t)$ and τ_x is the spatial shift variable. The detector at P_4 time integrates the light distribution in (2.15) for T seconds and produces the correlation output

$$\begin{aligned} u_4'(\tau_x, n) &= (1/T) \int_0^T s(t - \tau/2) s_n(t + \tau/2 - \tau_x) dt \\ &= s(t) \otimes s_n(t) = R_n(\tau_x). \end{aligned} \quad (2.16)$$

The top row of the filter mask at P_3 is transparent and thus along the top line in P_4 we find the wide-band correlation of the wide-band unfiltered signals s_a and s_b

$$R_T = (1/T) \int_0^T s(t - \tau/2) s(t + \tau/2 - \tau_x) dt. \quad (2.17)$$

Thus, as before, R_T provides narrow correlation peak widths and thus good time delay $\tau_x = \tau_m$ or angle resolution τ_m of the target, whereas the R_n correlations contain the necessary frequency distribution information on the target. We discuss these systems later in Sect. 2.10 and for now note that the narrow-band and wide-band signal correlations must be considered in an APAR adaptive in both space and time.

2.7 ELECTRONIC SUPPORT SYSTEM

In this section, we describe the two electronic support systems being assembled to provide the necessary drive and synchronization for the COC APAR correlators. We refer to the two systems as computer-driven and hardware electronic-support systems respectively. Since the received signal at one element of a phased array is the sum of the signals from N different noise sources each of which can be at a different angle, range, strength, center frequency and bandwidth, a single received signal is quite complex. Since each of its constituent component subsignals (from the different individual noise sources) changes with the noise source scenario and with the antenna spacings, producing the different received signals for each case is in itself a major effort. We intend to use a combination of simulation and experimentation (with the two electronic support systems) to conduct our APAR study. In this section, we consider the different electronic support systems, the advantages and disadvantages of each, the design and performance of each. An overview of both techniques is given to convey the key points of the diverse signal generation concepts.

In the computer-driven system, the two received signals are digitally computed using the simulator (Sect. 2.5) and their time histories are produced, stored in a digital file and recorded on tape. This magnetic tape is then brought from the main computer facility to our own dedicated PDP-11 where it is loaded into our tape file and then placed in the large 2 M-bit on-line memory peripheral available on our system. The signals are then read from this memory as base band data, D/A converted, up-converted to the center frequency of the AO cells by heterodyning with an oscillator using two mixers. Bias is then added through T splitters, the signals are then attenuated (ATT) to bring them to the proper range for the final power amplifier (AMP) to drive the point source (LED or AO modulator) and the AO line itself. This system is shown in block diagram form in Fig. 2.13.

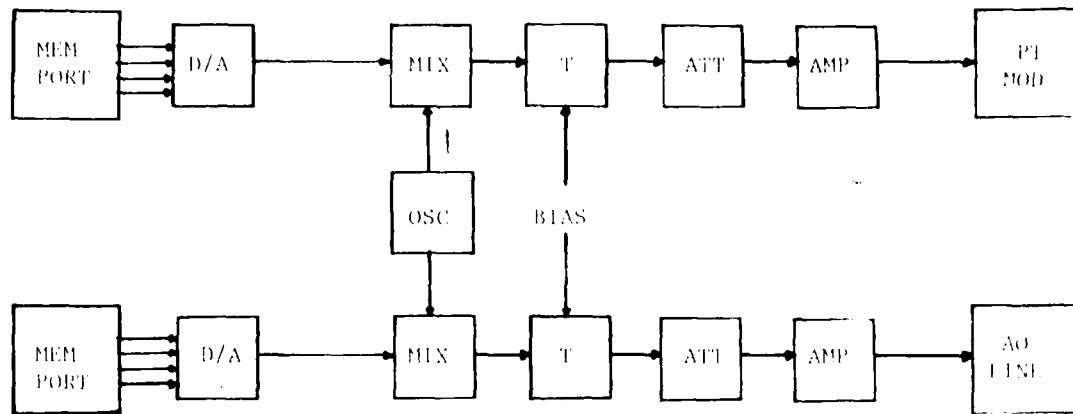


Figure 2.13 Block diagram of the computer driven electronic support system.

In the hardware system (Fig. 2.14), separate noise generators (NG) are used (one for each noise source). The bandwidth and amplitude of each can be controlled directly on the generators. Delay lines are used to delay and band pass filter each signal. Separate oscillators enable the center frequency of each noise source to be independently controlled. The signals from the different noise sources with the proper delays and bandwidths and center frequencies are then combined and fed to the standard T, ATT, AMP circuits to provide input signals of the necessary power and linearity for the point modulator on the AO lines.

From the above two paragraphs, the basic advantages and disadvantages of the two techniques should be apparent. The computer-driven system is capable of producing essentially any type of signal and is thus much more flexible than the hardware system. It also offers a larger range of delays than one can achieve with

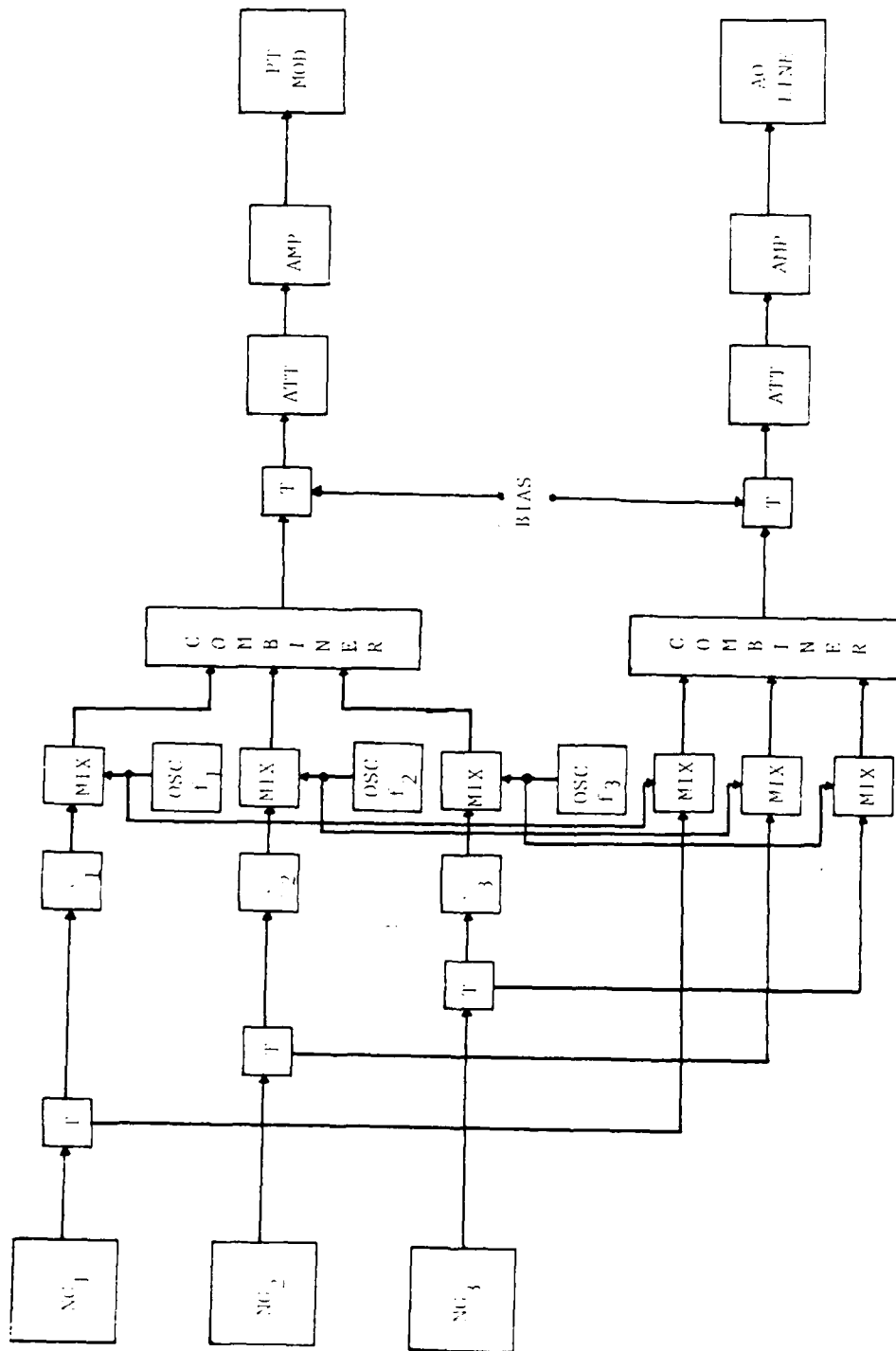


Figure 2.14 Block diagram of the hardware electronic support system.

hardware delay lines, probably better synchronization, and moreover the ability to accurately repeat long (32 msec) duration signals. This system is more complex in the sense that it requires operation of the entire PDP-11 system and its peripherals, but moreover the dynamic range and bandwidth of the data it can produce are severely limited by the speed and size of the on-line memory. Conversely, the hardware support system is severely limited in the flexibility of the signals and delays it can produce, but the system is more directly controllable and can produce larger dynamic range data. The bandwidth and duration of the signals can be longer in the hardware system, but random signals cannot be accurately repeated on this system. Inclusion of LFM, PRN and other deterministic signal sources in the system is possible and would improve its performance at the cost of addition expense and complexity. Both signal generation electronic support systems are thus seen to be mutually compatible and of use for different cases.

Let us now first quantify the performance of the computer-driven system. The on-line memory is the major limitation in this system. It is a 2-M-bit memory, normally configured as $512 \times 512 \times 8$ -bits. Its entire contents can be randomly accessed in 30 msec. Serial mode read out is normally used with one 8-bit word accessed about every 100 nsec. Thus, the minimum delay increment possible is $t_{\min} = 100$ nsec and delay increments in multiples of this are possible (there is no point in delays greater than 0.5 times the duration of the AO cell, or in our case 25 μ sec). Similarly, signal durations up to 30 msec (and approximately 250,000 samples) are possible and they can be accurately repeated by recycling the memory.

The bandwidth and dynamic range of the data from this system are its major limitations. With one 8-bit output word every 100 nsec, the signal bandwidth possible is 10 MHz (base band) or 20 MHz DSB. Two signals are necessary s_a and s_b . If these are real, then 4-bit words (16 gray levels) can be used for each, with four of the eight output bits at each sample corresponding to one 4-bit sample of

each signal. This is the situation shown by the two memory ports in Fig. 2.14. If complex-valued data is to be processed, then two signals (real and imaginary) are necessary for each waveform and four output signals must be obtained from each 8-bit output word from the memory (i.e. 2-bit or four level accuracy is possible). If less bandwidth is acceptable, 256 level or 8-bit samples can be obtained at 5 MHz by assigning each alternate output word to be a sample of a different signal.

These specifications are quite compatible with the available AO cell and the intended COC TSIAPAR processor. Moreover, a separate analysis has been performed by us that indicates that 2-4 bit quantization of random signal data does not appreciably effect the SNR of the resultant correlation, even when reasonable input noise is present [12]. Thus, this computer-driven system is quite attractive for the intended application. It is clearly the most flexible system capable of providing the widest range of waveforms and delays.

It is possible to extend both electronic support systems to provide the necessary input signals for complex-correlations. To achieve this, four signals are necessary from the memory, each pair is quadrature modulated and fed to the point modulator and the AO cell respectively. Although complex correlations yield 3 dB more SNR than do real correlations, in many adaptive radars, the phase of the input signal might not be preserved with heterodyning and thus complex correlation would not be of use. In practice, one can achieve complex correlation by inputting different signals and cycling the system twice and adding the separate outputs electronically with quadrature detection. Thus, complex correlations can be performed on the existing system with added complexity. Since the central point and purpose of the COC correlator can be proven without complex correlations, we have elected to perform real rather than complex operations on this system.

The computer-driven electronic support system is nearly operational. Additional funds are necessary for a new operating disc system and for redesign of the sync board to suppress horizontal and vertical blanking pulses during fly back. These are expected within several months. Increasing the bandwidth of the memory to 20 MHz is possible but would require extensive rewiring (twisted pairs) and additional high bandwidth clock, driver and receiver circuits plus higher frequency D/A converters. These improvements are possible but again are not necessary to prove the feasibility of the COC concept. All of the associated mixers, T connectors oscillators and amplifiers have been unified and are common to both electronic support systems.

For the above reasons, for near-term experiments on the COC APAR radar processors we will use the hardwired electronic support system of Fig. 2.14. The use of real not complex correlations is quite attractive for the hardware system since absolute synchronization of the real and imaginary parts of the signal are not needed then. The range of the delays possible in the hardware system is less than in the computer-driven one (400 nsec minimum), but this is still quite acceptable for APAR applications. By varying the delay line taps, incremental delays in units of 400 nsec can be realized. At 400 nsec, the clock to the delay line permits a maximum signal bandwidth of 2.5 MHz. It is possible to continuously vary the delay but as the delay is linearly increased, the bandwidth is linearly decreased. This clocking feature of the delay lines will be used to vary the bandwidth of the different noise sources in our electronic system.

In the system of Fig. 2.14, the frequency of the different noise sources (maximum of three) is set by separate oscillators. The bandwidth of each source will be controlled by varying the clock frequency to the delay lines. The delays will be adjusted with different tap weights on the delay lines. A low pass filter follows

each delay line to remove aliasing and to reconstruct the continuous bandwidth signal from the discrete one. This filter is chosen to cover the necessary source bandwidth range anticipated. The strength of each noise source is adjusted directly on the noise generator. A combiner (microwave splitter) is necessary rather than an operational amplifier to combine the signals because of the high frequencies involved.

The dynamic range of the electronic system is excellent, 60 dB, limited by the delay lines, compared to 23.5 dB for the two-channel 4-bit computer system. The oscillator used (Techtronics 5 G 503) operates from 250 KHz-250 MHz. The mixer is the Mini-Circuits model ZAY-3 with frequency response from 70 KHz-200 MHz and linear operation up to 15 dBm signal input. The Mini-Circuits model ZAY-3 attenuator has a frequency range from 1-200 MHz and continuously controllable attenuation. The broad band amplifiers are the EMI model 300 L with an operating range of 250 KHz-110 MHz. Their maximum RF output (3 W) is much larger than the 0.1 W needed for linear operation of the AO cells, thus their linearity at the operating power levels used is much better than the 2.5% obtained at 3 W.

The hardware electronic support system is fully operational. The computer-driven system represents a new feature emphasizing flexibility and multi-purpose operation of optical signal processing systems. Both systems are necessary to fully pursue a COC approach to APAR radar. Use of these and other support hardware in simulation systems are included in Sects. 2.9 and 2.10.

2.8 POST PROCESSING

In this section, we discuss various post-processing techniques by which the adaptive weights for the APAR can be obtained from data in other forms. The IOP (Chapter 3) and the WDP (Chapter 4) techniques operate on the covariance matrix M

of the antenna and thus can also be thought of as optical post-processing, whereas calculation of M (by digital or other techniques) is the primary preprocessing in this case. We refer to this method as sampled matrix inversion. Obtaining an accurate estimate of M must be given some attention. In [13], $K \geq 2N$ samples (for an N element array) were found to give an average loss ratio better than 3 dB in the sampled estimate of M . However, more recent work [14] has pointed out that this implies that one half of the time the loss will be above 3 dB and half of the time it will be below 3 dB. Thus, recent work suggests that $K \geq 4N$ be used and shows that this insures that the probability of the loss being above 3 dB is then only 0.0032. More attention to this issue of accurate estimation of the sampled covariance matrix is necessary and should be pursued in a later phase of this work.

Our major concern in post-processing arises in conjunction with the COC system. In this system, the location (in angle or time delay and frequency) of the noise sources is calculated in the optical system and the adaptive weights W must still be obtained from these data in a digital post-processor. The weights are chosen to produce nulls in the antenna pattern at noise locations while maintaining response in the signal direction, with no stipulations on the antenna response at other locations.

From the preliminary discussion in Sect. 2.6, we saw that the cases of narrow-band and wide-band noise and noise at the same angle and different frequencies and noise sources at the same frequency but different angles must be treated separately with different post-processing necessary (as described below) and with different slit filter widths h_n in Fig. 2.10 chosen for each case. In Sect. 2.10, we will unify these different cases. For now we first consider the case of a narrow-band noise source at an angle θ_m and at a frequency f_m . A simple post-processor can be used in this case as we now describe.

The time delay τ in units of seconds (the quantity measured in the COC processor) is

$$\tau = (d/c) \sin \theta_m \quad (2.18a)$$

and is thus independent of the frequency f_m of the noise source. However, the time delay in radians

$$\tau = \left(2\pi f_m \right) (d/c) \sin \theta_m \quad (2.18b)$$

depends upon the frequency f_m of the noise source. Moreso, when the adaptive weights w_n are computed, then the resultant antenna response $E(\theta)$ has a simple Fourier transform relationship

$$E(\theta) = \sum_n w_n \exp \left(j 2\pi f_m \sin \theta_m \right). \quad (2.19)$$

Thus, a simple digital FFT post-processor can be used. However, note that the frequency f_m of the noise source must be known in (2.19). Moreso, if the noise sources exist at different frequencies, then a different FFT is necessary to compute the weights w_n to place nulls at each (f_m, θ_m) parameter pair. If this is done separately for each noise source, no convenient way to weight the sum of the individual weights w_n to null each noise source has yet been found. Since one set of weights may place a null in the desired location, whereas another set of weights may place a peak there, the resultant response in the desired location will be non-zero.

Thus, we first consider the problem of the post-processing required when the noise sources lie at the same frequency and are narrow-band. We then consider the case of wide-band noise sources separately later. Each case will be shown to require a different post-processing algorithm. Solutions for each of the different noise pattern cases will be shown to exist (here and in summary in Sect. 2.10).

Thus, all APAR cases can be solved. From the form of the output data at P_4 of Fig.

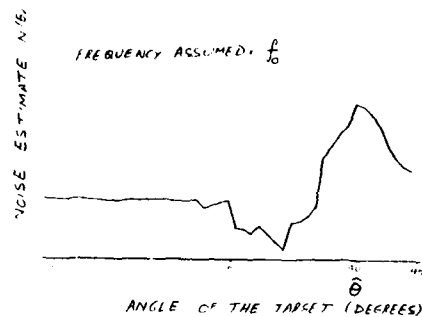
2.12, the post-processing necessary can be determined from the noise field present at a given time. Thus, the resultant APAR processor appears to be applicable to a variety of different cases.

Let us first consider the narrow-band noise case, when all noise sources are at the same frequency. From (2.18b), we see that noise sources at different frequencies f_m and angles θ_m can result in the same time delay τ_m in units of radians. This implies that if we chose an arbitrary frequency f_0 at which we assume the noise source to exist, the angle $\hat{\theta}$ computed for the noise source will satisfy the equation

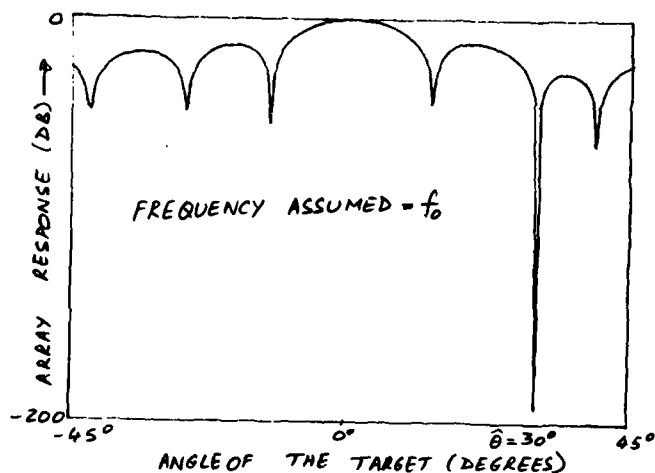
$$f_0 \sin \hat{\theta} = f_m \sin \theta_m, \quad (2.20)$$

where f_m and θ_m represent the correct frequency and angle of the noise source. This result in (2.20) is important because it implies that we can choose an arbitrary frequency f_0 at which to assume the noise source exists, calculate the weights w_n to produce a null at the angle $\hat{\theta}$ and that the resultant set of weights will automatically produce a null at the correct frequency f_m and angle θ_m as in (2.20).

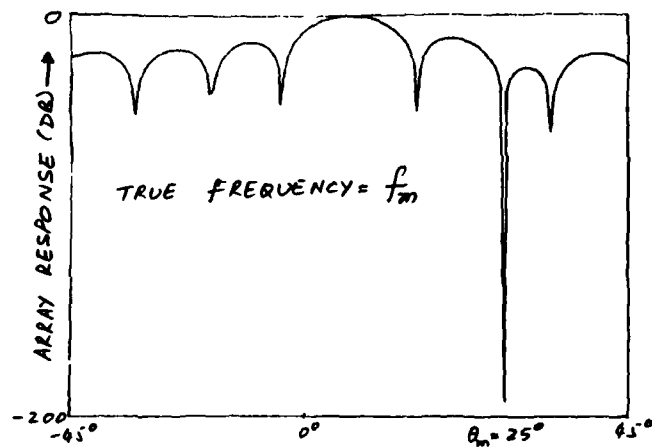
We refer to this adaptive array post-processing algorithm as the self-compensating or self-correcting algorithm. It is only of use with narrow-band sources and sources at the same frequency or at only a few different frequencies. To demonstrate the use of this algorithm, we simulated the multi-channel COC system for the case of a noise source at angle θ_m and frequency f_m . The multi-channel output from the COC system was obtained and slit integration of the pattern was performed as described in [2]. The resultant output shown in Fig. 2.15a exhibits a peak at $\hat{\theta}$, corresponding to a time delay τ_m calculated from (2.18a) that is independent of the frequency of the noise source.



(a)



(b)



(c)

Figure 2.15 Self-compensating adaptive antenna post-processing (a) $N_m(\theta_m)$ pattern, (b) null pattern assuming f_0 , and (c) assuming f_m .

The simulator (section 2.5) was then used to compute w_n for this $N_m \left(\begin{smallmatrix} \cdot \\ m \end{smallmatrix} \right)$ pattern and to then calculate $E(\theta)$ from (2.19) using an FFT routine. For an assumed frequency f_0 , the resultant $E(\theta)$ pattern of Fig. 2.15b was obtained. It exhibits a null at $\theta = \hat{\theta}$ as is desired. We then repeated the simulations for a noise source frequency f_m and obtained the same set of weights but the different $E(\theta)$ pattern shown in Fig. 2.15c. As seen this pattern exhibits a null at the desired θ_m location rather than at $\hat{\theta}$.

From this discussion and simulation example, we see that the self-correcting post-processing technique works. We also see other features and uses of the simulator for calculating the adaptive weights and the resultant far-field noise distribution.

We now consider a new algorithm by which to calculate the set of adaptive weights for the case of many noise sources at the same frequency but different angular locations. This technique uses a new version of a projection mapping (PM) technique [16] that has been used thusfar only for image restoration [17]. We apply this technique to the problem of calculating the adaptive weights for a APAR and we suggest a modified algorithm that results in much faster convergence of this routine. Such a technique is necessary to make the speed of the post-processor compatible with the high speed of the COC system that estimates the noise field. A digital post-processor solution is envisioned, although modifications to the IOP (Chapter 3) or the WDP (Chapter 4) system may make an optical version of this algorithm possible.

To describe the new PM algorithm, we reformulate the APAR problem as below. We consider an array of N receiving elements at vector locations $p_1 - p_N$ in 3-D space. We denote the unit vector in the direction of the signal by \underline{u}_2 . We describe the noise field by M unit vectors \underline{u}_i (where $i = 1 \dots M$) in the direction

of the M noise sources. Our problem is to compute the adaptive weights \underline{W} such that we maintain full response in the direction of the signal \underline{u}_s , i.e.

$$\underline{C}^T \underline{W} = 1, \quad (2.21)$$

and null the response in the \underline{u}_i directions, i.e.

$$\underline{C}_i^T \underline{W} = 0, \quad (2.22)$$

where $i = 1 \dots M$ and where the vectors \underline{C}^T and \underline{C}_i^T describe the signal and noise response as

$$\underline{C}^T = \left[\exp \left[j 2\pi \left(f_0/c \right) \left(\underline{p}_1 \cdot \underline{u}_s \right) \right] \dots \exp \left[j 2\pi \left(f_0/c \right) \left(\underline{p}_N \cdot \underline{u}_s \right) \right] \right] \quad (2.23)$$

$$\underline{C}_i^T = \left[\exp \left[j 2\pi \left(f_0/c \right) \left(\underline{p}_1 \cdot \underline{u}_i \right) \right] \dots \exp \left[j 2\pi \left(f_0/c \right) \left(\underline{p}_N \cdot \underline{u}_i \right) \right] \right] \quad (2.24)$$

Although the present formulation assumes all narrow-band noise signals to be at the same center frequency f_0 , the proposed solution can later be extended to the case of non-equal noise source frequencies using techniques in [19-20].

Eq. (2.22) gives $M + 1$ constraints on the solutions for the N weights \underline{W} . If $M = N - 1$, a unique solution exists

$$\underline{W} = \underline{A}^{-1} \underline{U}, \quad (2.25)$$

where \underline{A} is an $N \times N$ matrix with rows $\underline{C}^T, \underline{C}_1^T \dots \underline{C}_M^T$, i.e.

$$\underline{A} = \begin{bmatrix} \underline{C}^T, \underline{C}_1^T, \dots, \underline{C}_M^T \end{bmatrix}, \quad (2.26)$$

and \underline{U}^T is the N element vector

$$\underline{U}^T = \begin{bmatrix} 1, 0, \dots, 0 \end{bmatrix}. \quad (2.27)$$

When $M > N - 1$, no solution exists. In most cases of interest, the number of noise sources M will be less than the number of adaptive elements N or $M = N - 1$ (or in practice only the M largest noise source will be used). In this case, the pseudo-inverse method [15] can be used and the solution is

$$\underline{W} = \underline{A}^T \underline{A}^{-1} \underline{A}^T \underline{U} \quad (2.28)$$

where \underline{A} is now an $(M + 1) \times N$ matrix.

Computing the weights can thus be reduced to the common problem of solving linear equations for which many solutions exist. However, most are computationally complex, especially for complex array geometries.

In the modified PM technique we propose, each of the $M + 1$ constraints in (2.22) is treated as a hyperplane in an N dimensional space whose axes are the weights w_n to be found. The problem is to find the intersection of all of these hyperplanes. The projection method begins with an initial guess \underline{W}_0 for the intersection point. We then project this point onto the first hyperplane and obtain a new point \underline{W}_1 which we then project onto the second hyperplane to obtain \underline{W}_2 , etc. until the vector point \underline{W}_{M+1} is obtained after projection onto the last $M + 1$ hyperplane. This completes the first iteration cycle. \underline{W}_{M+1} is then orthogonally projected back onto the first hyperplane, giving \underline{W}_{M+2} and the process is continued until \underline{W}_{2M+2} is obtained at the end of the second iteration.

The procedure is repeated until convergence occurs. The speed of convergence is improved if the hyperplanes are orthogonal and the larger the angle between adjacent hyperplanes (closer to 90°) the faster the convergence [18]. To speed up convergence, we suggest a pair wise orthogonalization technique that makes adjacent hyperplanes orthogonal. The method proposed is similar to one used in image restoration [17]. We wish to implement both the original and the fast PM techniques and to

apply them to the APAR optimal weight problem. The modified PM technique should produce deeper nulls at the correct locations in less iterations than did the original technique.

The conventional PM technique requires MN^2 initial complex operations and M^2 complex operations per iteration. Conversely, the modified PM technique requires only $2NM$ initial operations and MN complex operations per iteration. Since $M \sim N$ in general, a large computational savings is obtained by the new second technique. We thus advocate such a digital post-processing method for computation of the adaptive weights in the COC system. When the frequencies of the N sources differ and when wide-band jammers are present, the technique in [19-20] should be used and incorporated into the modified PM technique. In these advanced methods, narrow-band noise sources can be assumed and the algorithm modified to produce wider nulls at the central frequencies of the noise sources. We hope to pursue such techniques in later phases of this program. For now we note that the modified PM technique can be used for multiple narrow-band sources and the advanced PM technique for wide-band noise sources. We also note that both techniques operate directly on the data obtainable from the COC output system in the form of the TSI processor of Fig. 2.12.

2.9 TI COC EXPERIMENTS

To obtain initial experimental results on the AO system of Fig. 2.1 in a new configuration and application, we considered the problem of the accuracy of an optical processing system for APAR. One approach to this problem is the use of residue arithmetic [2, 21, 22]. In such a system, the input data to be operated upon is presented as its residue values in a chosen basis set. Operation on such data is attractive for optical systems because the dynamic range of the bit information is restricted to the corresponding residue value and moreover because the associated data is in the format in which additions, multiplications, and subtractions are possible

without carries. The major problem for such a data representation is how to convert input information into a residue number system representation and the associated formatting required.

To achieve such operation at high data rates, we have devised a version of the TI AO system of Fig. 2.1 that produces the associated conversion needed in a time pulse position coding scheme rather than the conventional spatial pulse position coding technique [21]. In residue arithmetic, we represent an integer J by the n -tuple set of remainder residues R_{m_i} with respect to the N integer moduli m_i .

The system we used to demonstrate the above concept is shown in Fig. 2.16 in schematic form. In this system, the temporal source modulation is

$$s_0(t) = \delta(t - J\Delta t), \quad (2.29)$$

where we represent the value of the data by the time of occurrence of the pulse. The AO cell at P_1 is uniformly illuminated with this light distribution. The AO cell has a transit time $m\Delta t$ and is fed with a signal consisting of pulses of width Δt and period $m\Delta t$ where m is the residual modulus.

We describe the pattern at P_1 by

$$s_1(x, t) = \frac{1}{n} \delta(x - nm\Delta t + t). \quad (2.30)$$

Leaving P_1 we find $s_0 \otimes s_1$, which is time integrated at P_3 to produce $s_0 \otimes s_1$ in the form

$$f(x_3) = \frac{1}{n} \delta \left[x_3 - (J - nm) \Delta t \right]. \quad (2.31)$$

From (2.31), we see that the spatial location of the output peak at P_3 is proportional to the desired residue R_m of J modulo m .

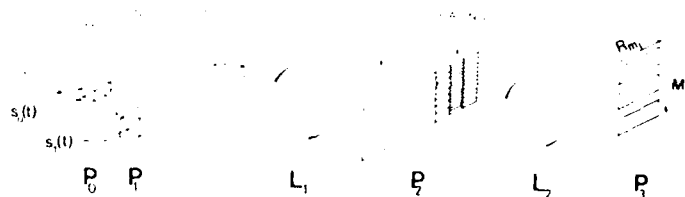


Figure 2.16 Schematic diagram of a TI AO correlator for residue arithmetic processing for APAR.

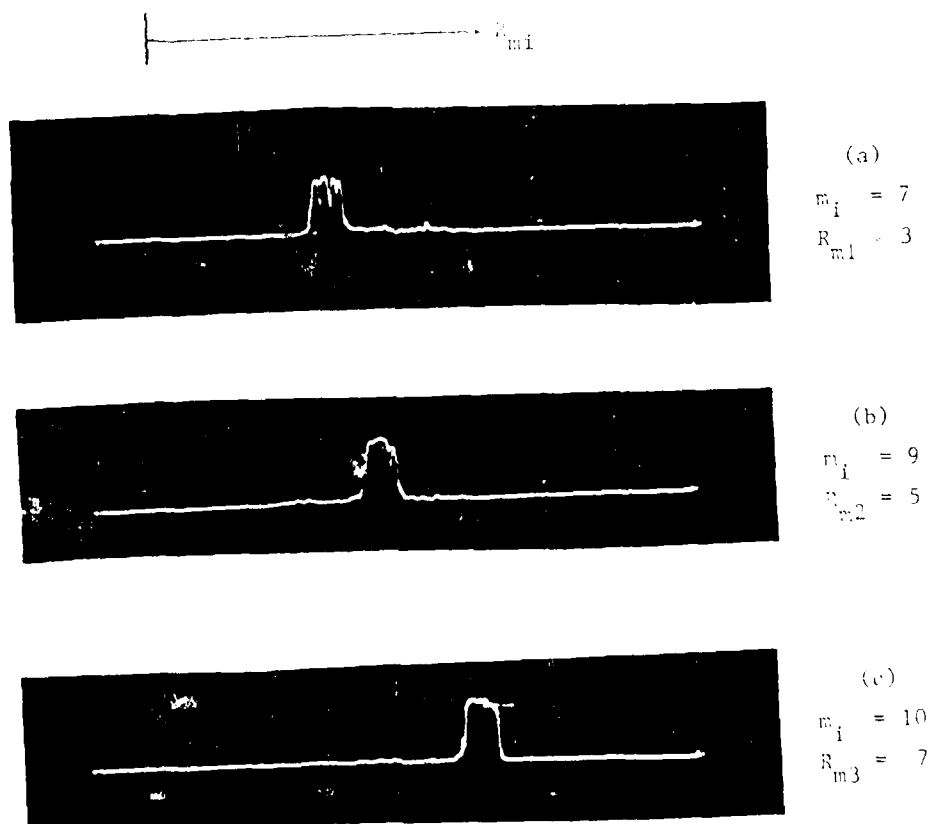


Figure 2.17 Output plane pattern from Fig. 2.14 for a decimal input $J \approx 437$ and moduli $m_1 = 7, 9$, and 10 .

To provide parallel conversion of J into R_{m_i} , for N different bases m_i , we modify $s_1(t)$ to be a sum of signals, each consisting of pulses of width Δt and different periods $T_i = m_i \Delta t$ and different associated carrier frequencies f_i within each Δt for each m_i . This causes each modulus to appear spatially separated at P_2 , where we block the dc and one first order term.

At P_2 , we place N vertical gratings. These cause the correlations $s_0 \otimes s_{1i}$ of s_0 with different modulo m_i signals to appear at different f_i spatial frequency locations in P_3 . The P_3 pattern is thus the N correlations $s_0 \otimes s_{1i}$ on N different horizontal lines. The horizontal position of each correlation corresponds to R_{m_i} with the m_i encoded in the vertical direction. Thus, the system of Fig. 2.16 provides parallel conversion of the input integer data J .

The system of Fig. 2.16 was assembled using an AO cell with $s_0 = 38$ MHz, bandwidth of ± 5 MHz, and $T_c = 32.25$ nsec. Moduli $m_i = 7, 9, 10$ were chosen with $\Delta t = 606$ nsec and associated periods T_i corresponding to pulse frequencies $f_i = 34.7, 38$, and 41.3 MHz. These correspond to spatial frequencies $f_{gi} = 55.9, 61.29$, and 66.61 . These result in a reasonable separation of 2.62 mm between gratings at P_2 . Appropriate grating spatial frequencies were chosen for use at P_2 (15, 35, 55 cy/mm).

The corresponding outputs are shown in Fig. 2.17. The position of the peak on each line corresponds to the residue R_{m_i} for the corresponding moduli m_i when evaluated.

An older AO cell of lower optical quality was used in these experiments. The final AO cells arrived too late in the program to permit the TI and APAR experiments to be performed on them. Similarly, equipment delays and lack of funds prohibited full use of the electronic support system until after completion of the contract period. These experiments still demonstrate the use of the AO cell and much of the

support electronics as well as the T1 AO correlator in a new application (residue arithmetic) addressing accurate optical data processing for APAR use. In the next section, the COC simulator is used with the TSI architecture to show computation of a far-field noise pattern adaptive in angle and frequency.

2.10 TSI EXPERIMENTS

In this section, we provide simulated verification of the TSI system of Fig. 2.12 and discuss the associated post-processing required to calculate the adaptive weights from its output data for diverse noise cases.

To demonstrate use of the TSI system of Fig. 2.12, two random uncorrelated noise sources with Gaussian statistics, zero-mean values and with variances $\sigma_1^2 = 1.0$ and $\sigma_2^2 = 1.4$, located at $\theta_1 = +23^\circ$ and $\theta_2 = -22^\circ$ with different bandwidths of 0.352" and 0.234" (2/3 of the first source) were produced using the simulator of Sect. 2.5. We let $-\pi$, 0, $+\pi$ in frequency or radian bandwidth correspond to samples 0, 256 and 512 respectively. The spectrum of the two noise sequences are shown in Fig. 2.18. For the first noise source, its bandwidth was 90 points out of 256 centered at the point $256 + 45 = 301$; whereas for the second noise source its 60 point bandwidth was centered at 286.

The angular locations of the noise sources were verified by performing the wide-band correlation corresponding to the top line in the output pattern of Fig. 2.12. The results are shown in Fig. 2.19. The relationship between the delay τ in radians and the angle θ of the noise source is

$$\tau = (2\pi d/\lambda) \sin \theta. \quad (2.32)$$

In the simulations, we assumed $d = 100 \lambda/2^\circ$ or

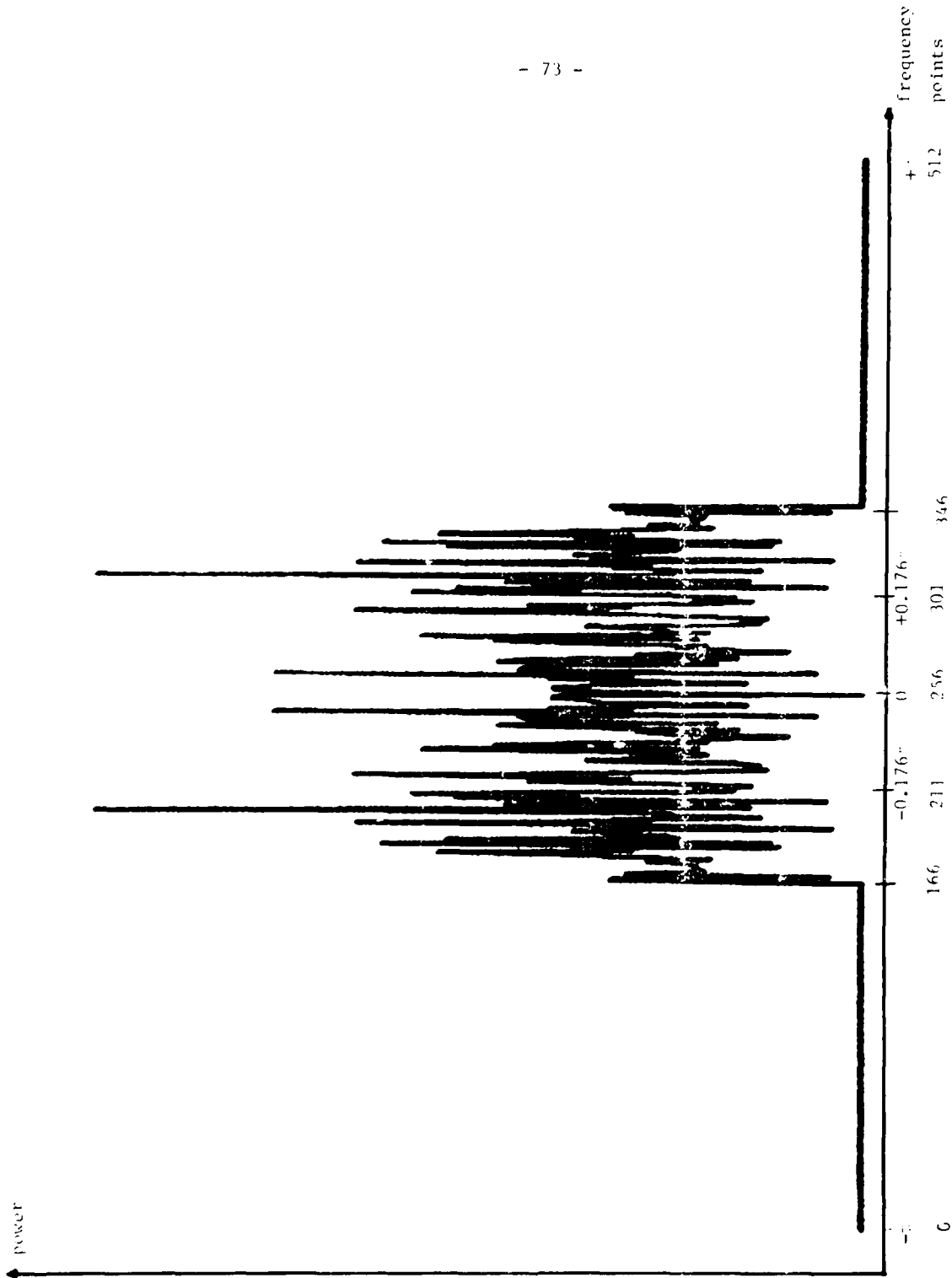


Figure 2.18a Power spectrum for the first noise source.

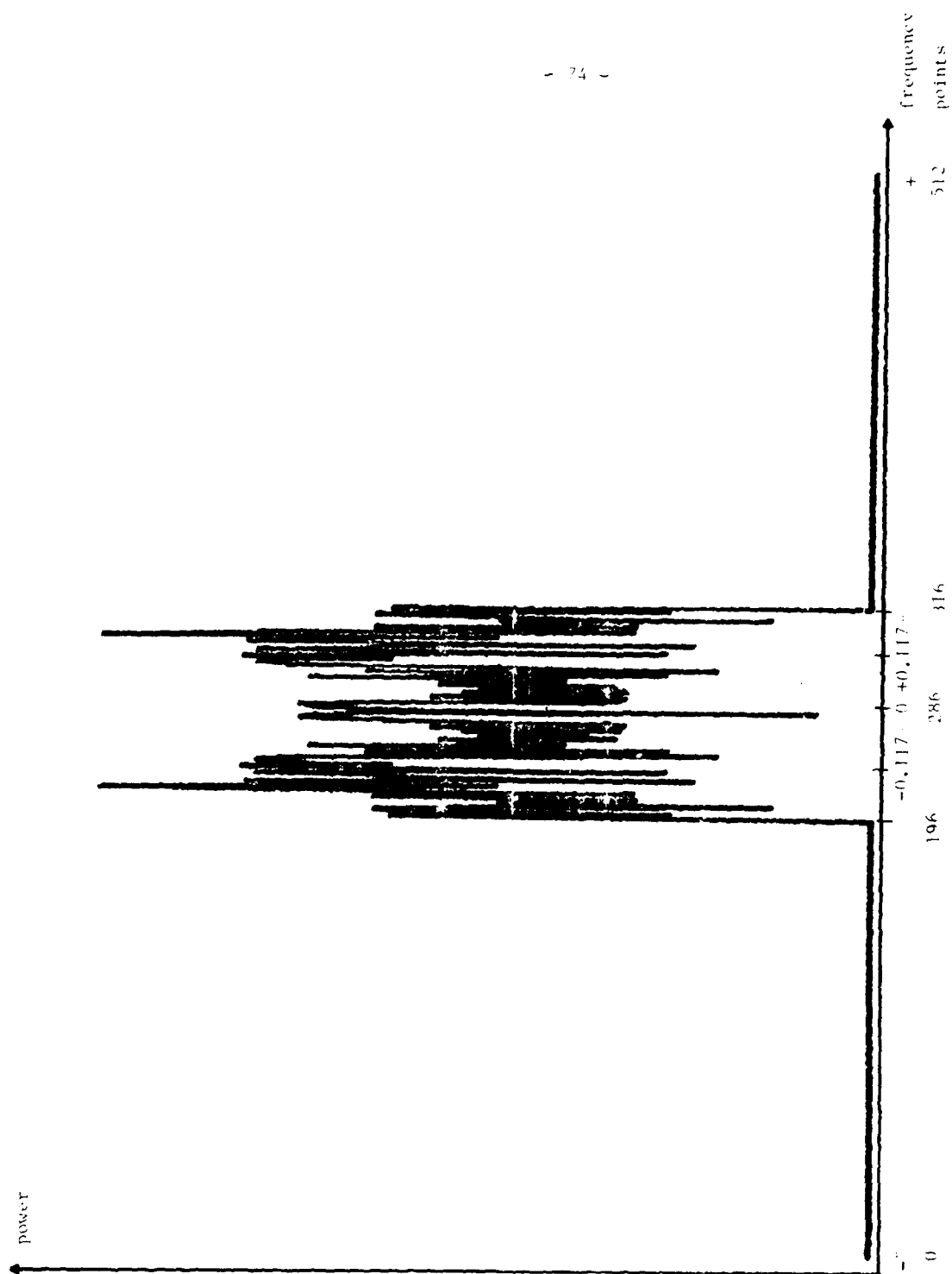


Figure 2.18b Power spectrum for the second noise source.

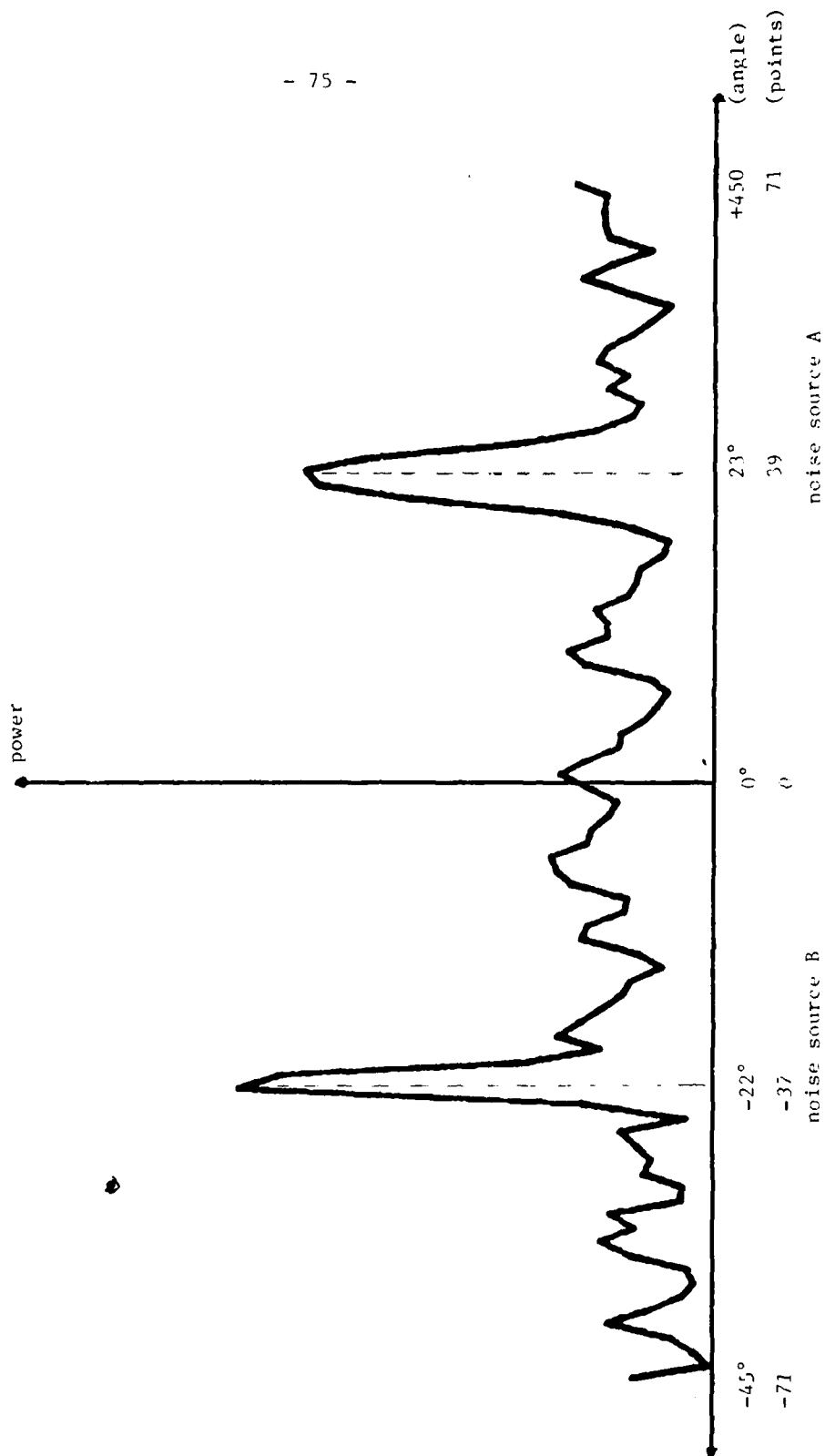


Figure 2.19 Wide-band correlation of two noise sources in the TSI system.

$$\tau = 100 \sin \theta \quad (2.33)$$

We employed a detector integration time $T = 512$ samples. The two noise source angles $\theta = +23^\circ$ and -22° correspond, using (2.33), to delays of +39 and -37 points respectively. The data in Fig. 2.19 confirm these noise source angle locations. The amplitude of the two noise peaks also agrees with the intended signal strengths with the second noise source (with the larger variance) producing the larger peak.

The $N = 9$ multi-channel outputs in Fig. 2.12 are shown in Fig. 2.20. The bandwidth of each band pass filter h was chosen to be 0.117π (30 points). This resulted in nine separate frequency filters. From Fig. 2.18 we see that the first source lies at 23° and has zero response beyond 0.352π ; whereas the second source lies at -22° and has zero response beyond 0.234π . These values are in exact agreement with the expected results. The strength of the response in the different frequency bands in Fig. 2.20 are approximately equal corresponding to the approximately flat spectrum (Fig. 2.18) of the noise sources over their bandwidth. Note the wideband (WB) output on the top line.

We now briefly consider the use of the TSI system (Sect. 2.6) and the various post-processing algorithms (Sect. 2.8) for different noise source cases in APAR.

For the case of a single source at one frequency, low correlation SNR is expected because of the zero bandwidth signal. We expect some signal bandwidth in practice and thus the TI correlator will yield adequate estimation of the target angle θ_m . Because of the self-compensating algorithm feature (Sect. 2.8) the noise frequency estimate is not of concern in this case. Multiple sources of the same frequency perform similarly and if their bandwidth is low, their correlation peaks will be wide and accurate discrimination of the different angles of the noise sources may be difficult. For such cases, beam forming techniques may be preferable.

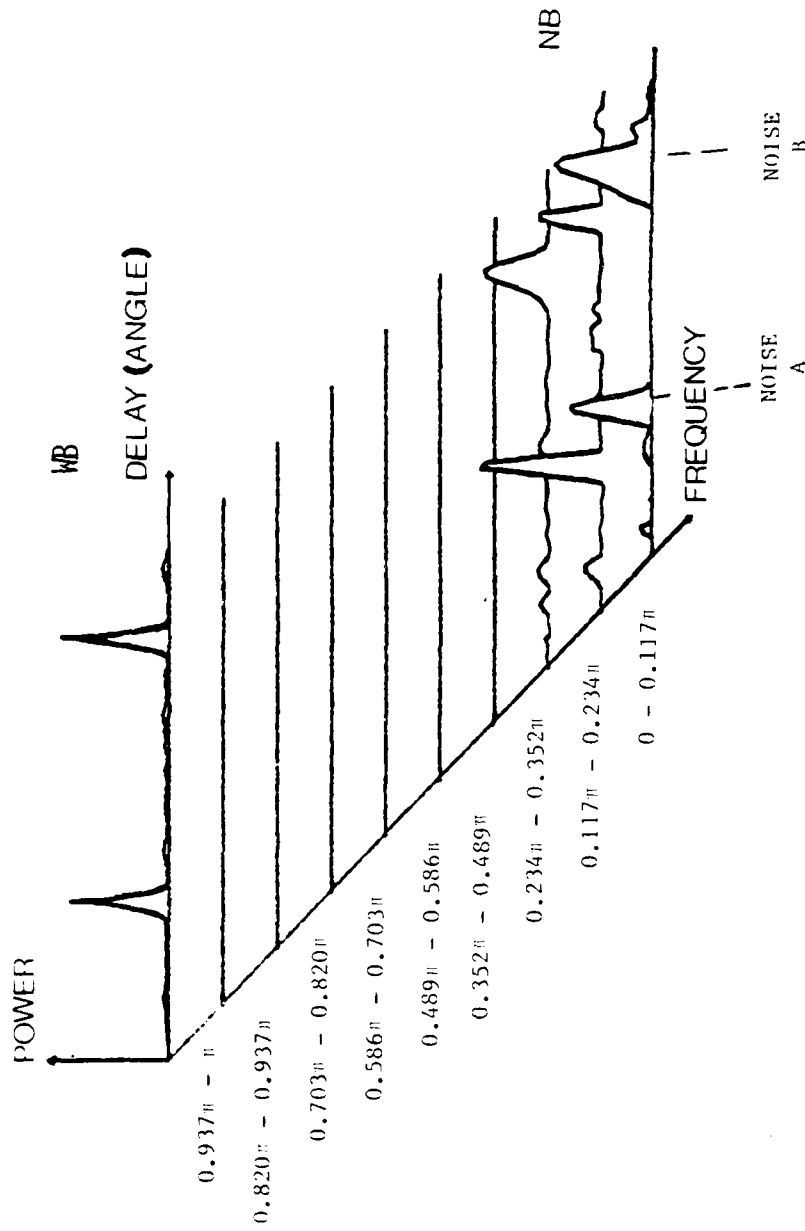


Figure 2.20 Multi-channel narrow-band correlation of two noise sources in the TSI system.

For the case of a single noise source with multiple frequencies, we have a wide-band jammer. In this case, the self-compensating feature is not appropriate. Instead, we use the TSI system and the modified projection method with additive features to produce a wider null at the center frequency (if the noise source frequencies are in one band) or the advanced projection technique in other cases.

Choices of the slit widths (or the effective bandwidth of the different filters) in the TSI mask and other issues such as the combined use of the wide-band correlation output and the multi-channel narrow-band correlations (the first to accurately determine the angular target locations and the combination to deconvolve the frequency response of each target) were described in Sect. 2.6.

2.11 SUMMARY AND CONCLUSION

Our new COC work on APAR processing has resulted in many new algorithms and system architectures as well as in the fabrication of many components for the COC processor. In the past one year we have:

- (1) Developed a new COC concept using 1-D AO cells rather than 2-D SLMs (because they are more easily fabricated and more readily available) and TI rather than SI correlators (because their longer integration times allow better noise statistical estimates) (Sect. 2.2).
- (2) Fabricated and performed initial testing of two AO cells (Sect. 2.3).
- (3) Developed a new simulator to handle multiple phased array signals with adaptivity in space and frequency, to compute weights and calculate corrected antenna patterns, plus conventional Fourier transform and correlation routines (Sect. 2.5).
- (4) Devised a new adjunct antenna concept that allows full space and frequency APAR data to be obtained using only a two-channel processor (Sect. 2.4).

- (5) Developed a new hybrid time and space integrating AO correlator architecture that combines the best features of the TI and the SI systems, and provides a 2-D display of the angle and frequency location of the far-field antenna noise pattern from the two-element adjunct antenna data (Sect. 2.6).
- (6) Designed and fabricated a hardware electronic support system and designed and nearly completed fabrication of a multi-purpose computer-driven electronic support system for complex noise source scenarios (Sect. 2.7).
- (7) Developed a self-correcting post-processing system, a new fast iterative modified projection method technique and an advanced projection concept to compute the adaptive weights for narrow-band and wide-band jammers that differ in both angle and frequency (Sect. 2.8).
- (8) Performed an experimental verification of a TI correlator and used it to demonstrate residue arithmetic computations in a new time position coding scheme. Residue arithmetic is an advanced technique whereby numerical computations can be performed in parallel with no carries and with reduced dynamic range requirements. This is necessary to realize an accurate optical computer (Sect. 2.9).
- (9) Demonstrated the use of the TSI system for computation of the 2-D space and frequency output antenna pattern from an adaptive array using an adjunct antenna (Sect. 2.10).

Thus, in conclusion, we have devised a new COC optical processor, demonstrated and simulated the key points of the system, begun development of the necessary post-processor, and fabricated the necessary components and support system to fully study this COC technique for APAR.

CHAPTER 3 ITERATIVE OPTICAL PROCESSOR (IOP)

3.1 INTRODUCTION

In this chapter, we describe our recent research on the IOP system. Since this system has been outlined in Chapter 1 and described in depth elsewhere [2-6], we concentrate on our recent progress on this system. In Sect. 3.2, we describe the new system design adopted for fabrication and in Sect. 3.3 we describe the system we fabricated during the past year. The basic and advanced versions of the IOP simulator and their use in APAR processing and in the analysis of the IOP are then described in Sect. 3.4. Several specific IOP operating issues such as accuracy, performance, corrections, convergence, convergence rate, selection of the acceleration factor, scaling of the eigen-values of M, etc. are discussed in Sect. 3.5. Initial experimental demonstrations of the IOP systems use in APAR applications are then presented in Sect. 3.6. A summary and our conclusions are then advanced in Sect. 3.7.

3.2 SYSTEM DESCRIPTION

3.2.1 Error Sources

In our prior work [2-6], the basic IOP system was developed and studied. Our analysis of this system showed three major error sources that would limit the accuracy and performance of the system. Fig. 3.1 shows the simplified IOP.

Imaging the LEDs vertically onto the mask without cross talk was found to be a severe problem as was uniformly illuminating each row of the mask with the light output from one LED. Our solution to these problems was to employ fiber optic coupling between P_1 and P_2 . We chose ten LED elements, a 10 x 10 mask and thus a 100

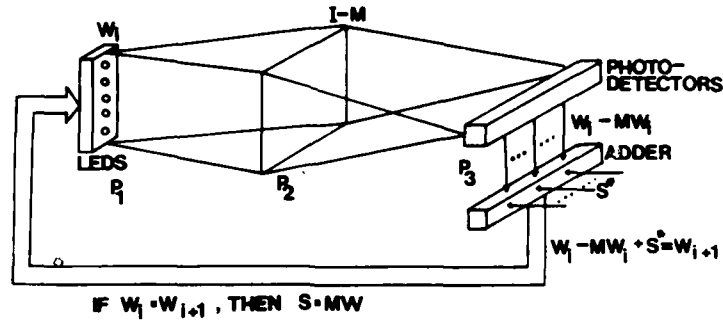


Figure 3.1 Schematic diagram of the iterative optical processor (IOP).

element fiber optic bundle. These numbers were chosen because such a system was of adequate size to allow it to be fabricated at reasonable cost and to be used in processing APAR problems of reasonable complexity and to allow unforeseen fabrication problems to emerge.

The non-linear response (current input versus light intensity out) of the LEDs tested made amplitude modulation unattractive without complex circuits to correct for the LED non-linearities. An investigation of laser diode sources showed that they had much better linearity over a very large dynamic range, but repeated attempts to obtain a linear array of these elements were unsuccessful. We thus chose to use pulse width modulation (PWM) of the LED sources.

With these major changes made in the system design (pulse width modulation and fiber optic interconnections), we then considered four other system component features that would affect performance of the final system.

Non-uniformity in the saturation level of the LEDs existed together with the non-uniform response of the linear photo diode detector. Non-uniform coupling of the LED sources to the fiber optic system and non-uniform outputs from the fiber optic array (due to polishing differences at the ends of the elements in the fiber optic bundles) were two other problems. We chose to correct for the non-uniform LED saturation levels and the vertical non-uniformity in the mask by a dynamic RAM in the electronic feedback system to adjust the gains of the different LEDs on-line. All four errors and their residual values (after RAM correction) were measured and can be corrected for by fabricating a fixed correction mask to be placed behind the adaptive mask B at plane P_2 in the system.

Electronic noise in the detector and associated support electronics was another problem of concern. We note that there are two components of this noise: a fixed signal pattern due to the dark current profile across the linear photo detector array and a time varying component due to other noise sources. We corrected for the fixed pattern detector noise by using a time-multiplex differential output scheme (see Sect. 3.2.4) in which a single complex multiplication is divided into two parts, each is run sequentially and the sequential outputs are subtracted.

The above system design and system operational changes provide a quite accurate IOP system for APAR use. Residual errors are still expected, depending upon the accuracy to which the corrections are fabricated and the magnitude of the time-varying noise sources in the system. We have measured this residual error and by simulation and lab verification determined its effect on the accuracy of the final \underline{W} estimate and the resultant null depth in the adaptive antenna pattern. These issues are described and quantified in Sects. 3.5 and 3.6. To prevent such error sources from causing divergence of the solution to the vector-matrix equation, we include a tolerance difference ϵ between two estimates w_i and w_{i+1} . When $\overline{w_i - w_{i+1}} < \epsilon$,

the iterative process is stopped. This prevents the system noise from causing the output to diverge from the correct answer as it is approached.

3.2.2 General Purpose Processor Considerations

This IOP system is a quite general purpose optical processor capable of solving any general vector-matrix equation of the appropriate form. Since optical systems are in general special purpose, this architecture is quite unique. To maintain the general purpose features of the IOP and to allow its use in other associated APAR tasks and in other applications, several design considerations were made. These choices were also made to decrease cost of the prototype system and the associated electronics. We have used demultiplexers (DMUX) at the LED inputs and the photo detector outputs. This decreases the throughput of the present system but greatly decreases the associated LED drive and photo detector hardware necessary. We have also only used a 1 MHz clock frequency for the LED sources. Although higher speed operation is possible (especially with laser diodes), the requirement on the associated digital and analog hardware become quite cost prohibitive for the available funding level. It is also possible to perform addition of the steering vector by appending an additional row to the covariance matrix mask and driving it with a separate linear LED array. Instead of this, we will perform this vector addition in a ALU (arithmetic logic unit) by digital multiplication in the electronic feedback loop. As noted earlier, a time-sequential system for handling complex data is also used to decrease the space bandwidth product (SBWP) requirements of the matrix and vectors in the system, thus enabling more complex problems to be addressed on the present system. Although the present system employs a fixed matrix mask transparency, it can later be replaced by an electroded spatial light modulator (SLM) and thus be made fully adaptive.

AD-A105 124

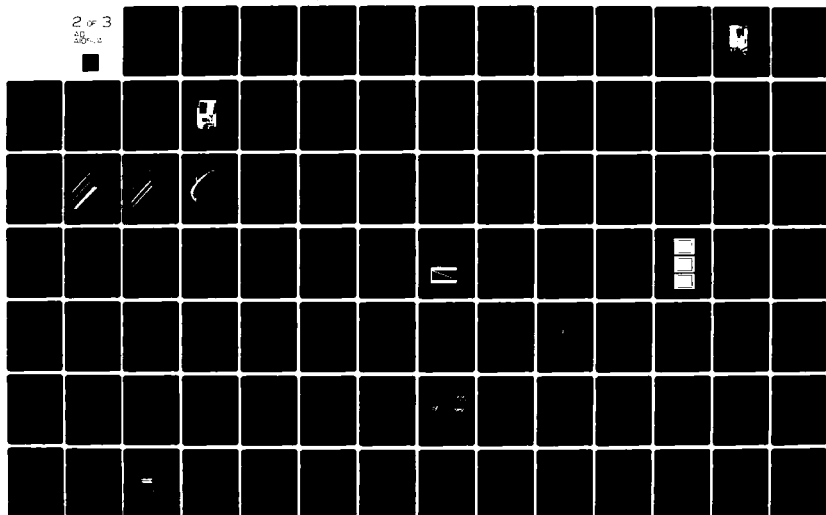
CARNEGIE-MELLON UNIV PITTSBURGH PA DEPT OF ELECTRICAL--ETC F/6 17/9
OPTICAL PROCESSING FOR ADAPTIVE PHASED ARRAY RADAR.(U)
JUN 81 D CASASANT F30602-80-C-0048

UNCLASSIFIED

RADC-TR-81-152

NL

2 of 3
10
30602



Thus, the present laboratory IOP prototype system represents a real-time optical processor for APAR of sufficient size, speed and complexity to enable it to be used for a quite diverse selection of signal processing problems. It also allows the accuracy and performance of the system to be assessed and new iterative algorithms to be incorporated without major hardware modifications.

3.2.3 Notation

The consistent notation to be used in describing this system is listed in Table 3.1. Underlined lower-case and upper-case letters denote vectors and matrices respectively. The letters a, B, and c refer to actual optical and electrical system parameters and are used to describe the amplitude of specific functions in the system. The letters x, H, and y are used to refer to general algebraic equations and operations. The elements of x, H, and y are bipolar, whereas for explicitly complex-valued data elements, we use s, M, and w. Various subscripts are used throughout. Their notation will be obvious in the different cases. The detector output is the actual vector-matrix product, but to this we add another vector to obtain the new input. For notational simplicity, we represent the detected vector-matrix output (after iteration *i*) by c_i and the vector to be added by c. This will simplify many vector-matrix equations throughout the text. The two uses of c should not cause difficulty when taken in context. We also denote versions of a at different iterations by a_i or a_k. A similar notation is used for the algebraic quantities x, H, and y.

3.2.4 Complex-Valued Data Handling

The steering and adaptive weight vectors s and w as well as the covariance matrix M, have complex-valued elements. Conversely, the LED outputs and the mask transmittance in the IOP are real and positive valued quantities. Thus, considerable

TABLE 3.1 IOP NOTATION

Parameter	System Parameters	Algebraic Parameters	
		(Bipolar)	(Complex)
Input Vector	\underline{a}	\underline{x}	\underline{w}
Input Vector Element	a_m	x_m	w_m
Matrix	\underline{B}	\underline{H}	\underline{M}
Matrix Element	b_{mn}	h_{mn}	M_{mn}
Detector Output (iter i)	\underline{c}_i	\underline{y}_i	\underline{s}_i
Detector Output Element	c_i	y_i	s_i
Added Vector	\underline{c}	\underline{y}	\underline{s}
Added Vector Element	c_n	y_n	s_n
Input Vector (iter i)	\underline{a}_i	\underline{x}_i	\underline{w}_i

attention is necessary to enable the IOP to accommodate the complex-valued data elements necessary for the APAR problem.

After studying various techniques by which to operate on complex-valued data in a real and positive IOP system and considering the static noise detector pattern, a sequential bipolar-data technique was chosen. To describe this technique, we first recall that the elements b_{mn} of the actual optical mask used at P_2 must satisfy $0 \leq b_{mn} \leq 1$ to be normalized. We choose to operate on bipolar complex-valued data by performing the operation

$$\begin{bmatrix} \underline{y}_r \\ \underline{y}_i \end{bmatrix} = \begin{bmatrix} \underline{H}_r & -\underline{H}_i \\ \underline{H}_i & \underline{H}_r \end{bmatrix} \begin{bmatrix} \underline{x}_r \\ \underline{x}_i \end{bmatrix} \quad (3.1)$$

in the system, where all quantities in (3.1) are bipolar and subscripts r and i denote the real and imaging parts of the corresponding vector or matrix.

As noted earlier, we use two sequential processor cycles to perform a complex-valued vector-matrix product. In the first cycle, the inputs are the positive parts, \underline{x}_r^+ and \underline{x}_i^+ , of \underline{x} and the outputs produced are the positive parts of \underline{y} , \underline{y}_r^+ and \underline{y}_i^+ . The same optical matrix \underline{B} is used for all vector-matrix operations until \underline{M} changes. Recall the matrix \underline{H} is related to \underline{M} by

$$\underline{H} = (\underline{I} - \underline{M}), \quad (3.2)$$

where \underline{I} is the identity matrix. We denote the minimum and maximum element values of \underline{H} by \underline{h} and \bar{h} respectively. The optical matrix \underline{B} used is thus generated from the matrix \underline{H} by

$$b_{mn} = \frac{\bar{h}m_{ii}}{\bar{h}-\underline{h}} - \frac{\underline{h}m_{nn}}{\underline{h}-\bar{h}}. \quad (3.3)$$

Subtraction of \underline{h} in (3.3) ensures that $b_{mn} > 0$, whereas dividing by $(\bar{h}-\underline{h})$ in (3.3) normalizes \underline{B} such that $0 \leq b_{mn} \leq 1$. The \underline{H} matrix to be used is thus $2M \times 2N$ in size with submatrices $\underline{M}_r, -\underline{M}_i, \underline{M}_i, \underline{M}_r$ from left to right and top to bottom as in (3.1). \underline{H} is obtained from \underline{M} by (3.2) and the corresponding optical matrix \underline{B} is obtained from \underline{H} by (3.3).

In the first and second cycles of the system, the processor performs the operations

$$\begin{bmatrix} \underline{y}_r^+ \\ \underline{y}_i^+ \end{bmatrix} = \begin{bmatrix} \underline{H}_r & -\underline{H}_i \\ \underline{H}_i & \underline{H}_r \end{bmatrix} \begin{bmatrix} \underline{x}_r^+ \\ \underline{x}_i^+ \end{bmatrix} \quad (3.4)$$

and

$$\begin{bmatrix} \underline{y}_r^- \\ \underline{y}_i^- \end{bmatrix} = \begin{bmatrix} \underline{H}_r & -\underline{H}_i \\ \underline{H}_i & \underline{H}_r \end{bmatrix} \begin{bmatrix} \underline{x}_r^- \\ \underline{x}_i^- \end{bmatrix} \quad (3.5)$$

respectively. The plus and minus bipolar components of the input vector \underline{a} are formed as

$$a_m^+ = (x_m^+ + |x_m^+|)/2 \quad (3.6a)$$

$$a_m^- = (x_m^- - |x_m^-|)/2 \quad (3.6b)$$

respectively, such that $a_m^+ = x_m^+$ if $x_m^+ > 0$ and $a_m^- = x_m^-$ if $x_m^- < 0$. The scaled and correctly biased (for input to the LED at the next cycle) output, after combining the system outputs from two successive cycles is then

$$y_n = (\bar{h}-\underline{h}) \left[\sum_m a_m^+ b_{mn} - \sum_m a_m^- b_{mn} \right] + \underline{h} \sum_m x_m \quad (3.7)$$

or in algebraic terms

$$\underline{y} = (\underline{h} - \underline{h}) [\underline{B} \underline{a}^+ - \underline{B} \underline{a}^-] + \underline{h} \underline{x}_m, \quad (3.8)$$

where the vector inputs at each of two sequential cycles are represented by \underline{a}^+ and \underline{a}^- , thus effectively realizing

$$\begin{bmatrix} \underline{y}_r \\ \underline{y}_i \end{bmatrix} = \begin{bmatrix} \underline{y}_r^+ \\ \underline{y}_i^+ \end{bmatrix} - \begin{bmatrix} \underline{y}_r^- \\ \underline{y}_i^- \end{bmatrix}. \quad (3.9)$$

The basic processor is thus a bipolar vector-matrix multiplier. It is described by the signal flow graph of Fig. 3.2. The necessary pre- and post-processing, scaling and normalization operations are performed in the micro-processor electronic feedback system (Sect. 3.2.5). Another technique for handling complex data is described in Chapter 4 in conjunction with the wavelength diversity processor (WDP) version of the IOP system.

3.2.5 Micro-Processor Feedback System

A micro-processor feedback system is necessary to perform the pre-processing of the vector \underline{x} and the matrix \underline{H} , the post-processing to perform the different operations in (3.7) and the associated scaling and biasing needed for the data. The micro-processor system also performs the necessary addition of the vectors \underline{c} , \underline{y} or \underline{s} to the bipolar vector-matrix product. It also controls the PAM correction circuits for the photo detector output and the LED input. The LED driver in the output demultiplexer and associated other system controls are also included in the electronic feedback system together with various storage and readout features that enable various vector-matrix products and iterative output products to be stored in digital memory, displayed sequentially on an oscilloscope in single-step or continuous mode, etc. Details of this aspect of the system are included in Sect. 3.3. A schematic of this hardware electronic feedback loop is included in Fig. 3.3.

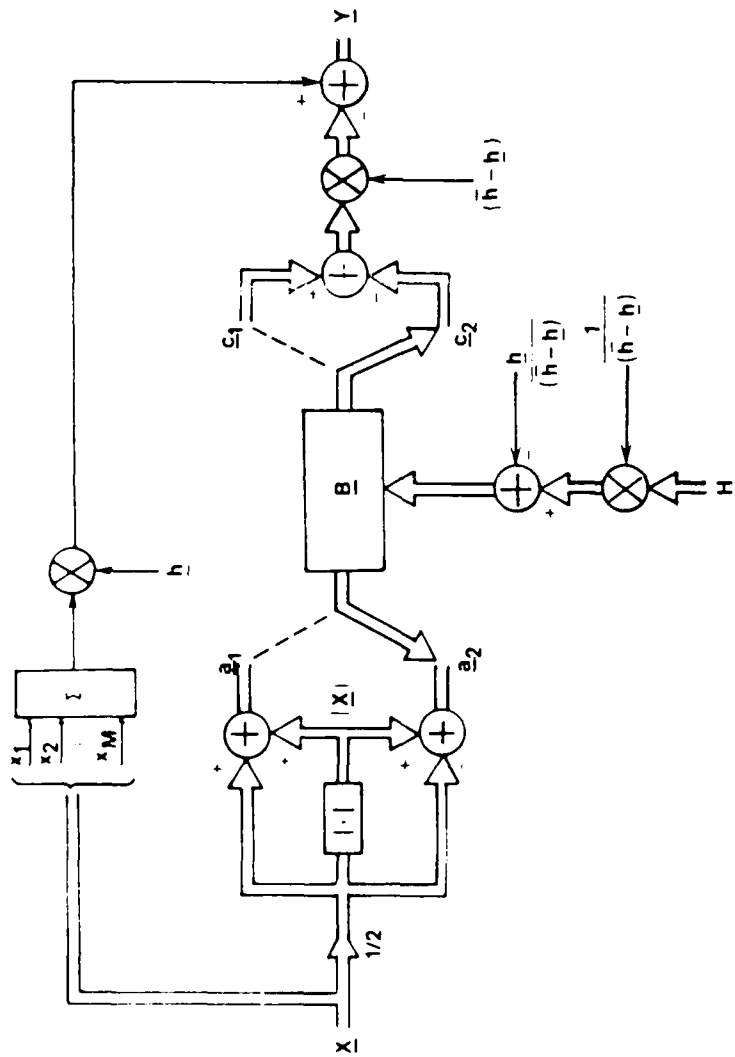
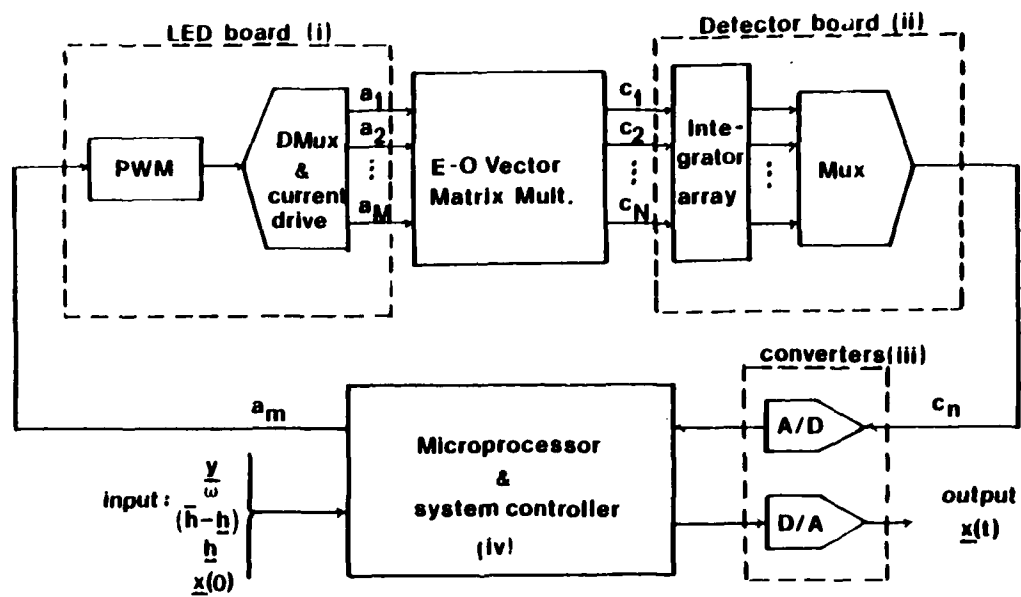


Figure 3.2 Signal flow diagram of the bipolar vector-matrix multiplier.



Iterative Optical Processor: Hardware Block Diagram

Figure 3.3 Schematic diagram of the hardware electronic feedback portion of the IOP.

3.3 SYSTEM FABRICATION

3.3.1 Optical Vector-Matrix Multiplier

A schematic diagram of the optical vector-matrix multiplier is shown in Fig. 3.4. The input consists of a linear array of 10 RCA SG-1002 LEDs which typically emit 1 mw for a 50 ma drive current at 940 nm. The LEDs are mounted on 0.15 inch centers along a copper block 1.5 inches long and held in place with silver epoxy. A fiber optic element manufactured to our specifications is used to distribute the light from each LED across the matrix mask. At one end it contains an array of 10 apertures, over which the LEDs are positioned. Each of these apertures contains a bundle of ten glass fibers, each 0.001 inch in diameter, the bundles are anchored in brass collars set in an Al block over the LEDs. The fibers branch outward to form a 2-D array of 10 x 10 or 100 fibers that are set in a stainless steel face-plate. The fibers are arranged in a 10 x 10 rectangular matrix with a center-to-center spacing of 0.014 inch vertically and 0.0375 inch horizontally between fibers. The fiber ends were polished and the entire unit potted in black latex in an Al casing. The optical matrix mask is sandwiched between the fiber optic element and the detector array as shown in the figure. The detector used is a Centronics LD-20 silicon photo diode array which contains 20 elements each measuring 4 x 0.9 mm on 0.95 mm (0.0374 inch) centers. The detector elements and the size of the matrix mask are chosen such that each detector element sums all of the light emerging from a column of the matrix mask. This provides the summation over the m rows in each column and produces the desired output

$$c_n = \sum_{m=1}^M a_m b_{mn}. \quad (3.10)$$

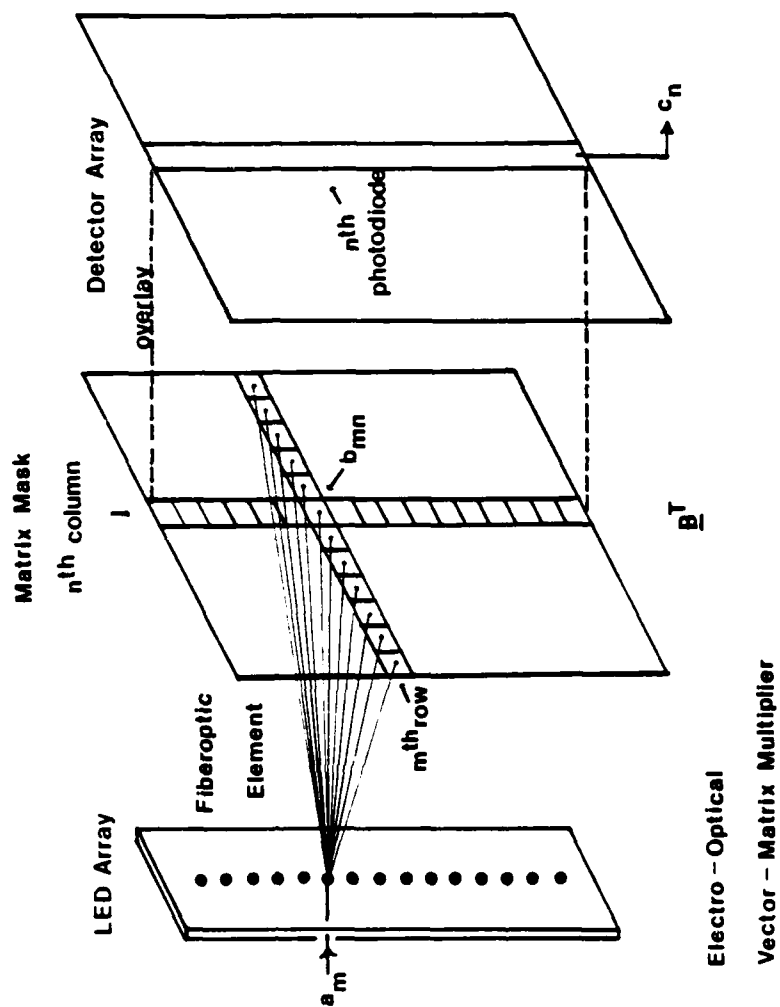


FIGURE 3.4
ELECTRO-OPTICAL VECTOR-MATRIX MULTIPLIER

A photograph of the optical vector-matrix multiplier is shown in Figure 3.5. Right to left are the LED array (sealed in white RTV compound) bolted to the fiber optic element, matrix mask and detector array. (The components are separated for clarity in the figure.) The fiber optic element, matrix mask and detector are mounted on mechanical translation stages and tilt positioners to facilitate alignment of all components.

3.3.2 Electronic Feedback System

The electronic feedback system is composed of the following four subsystems:

- (i) LED pulse width modulator (PWM), demultiplexer (DMUX) and current drive,
- (ii) time-integrating photodetector and multiplexer (MUX),
- (iii) D/A and A/D converters and
- (iv) microprocessor controller.

These subsystems and their interconnections are shown in the block diagram of Figure 3.6.

The LED board [subsystem (i)] consists of a clock (reference frequency, f_0), digital PWM, current drive and DMUX. Each element of the non-negative input vector $\underline{a} = \{a_m\}$ is fed (by the microprocessor) in digital form to the LED board and is converted by the PWM and current drive to a current pulse of duration a_m/f_0 . The amplitude of the pulse is fixed so that the peak power, P_m , radiated by the m -th LED is constant and its energy varies linearly with a_m . Amplitude modulation of the drive current would result in nonlinear distortion due to the exponential (current-power) response characteristic of the diode. The time-varying output from the m -th LED is therefore:

$$I_m(t) = P_m \text{ Rect}(f_0 t/a_m) \quad (3.11)$$

and the energy it provides is $P_m a_m/f_0$. All $M=10$ LEDs are selected in turn by the DMUX and are pulsed in this manner. The time T required to enter the vector \underline{a} is data dependent and equals

$$T = \frac{1}{f_0} \sum_{m=1}^{M=10} a_m. \quad (3.12)$$

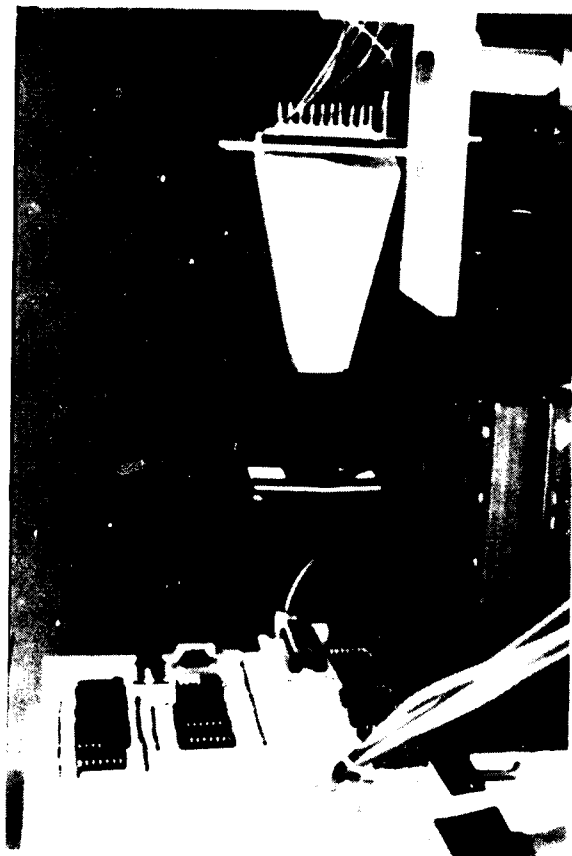
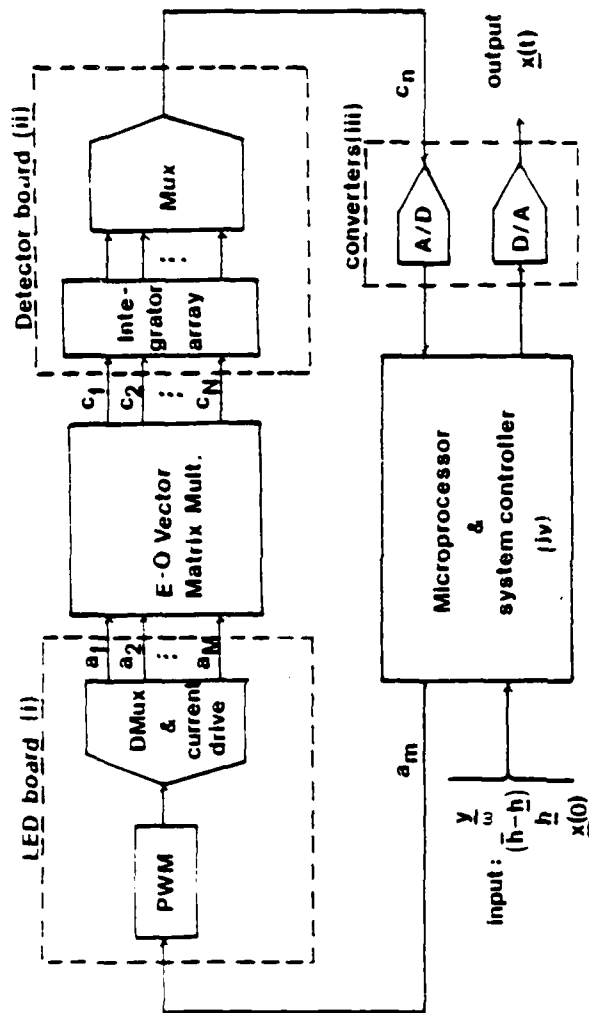


FIGURE 3.5
PERSPECTIVE VIEW OF THE LED ARRAY, FIBER OPTIC ELEMENT AND DETECTOR



Iterative Optical Processor: Hardware Block Diagram

FIGURE 3.6

ITERATIVE OPTICAL PROCESSOR: HARDWARE BLOCK DIAGRAM

Complex data is handled by separating the input vector into its positive and negative real and imaginary components and time-multiplexing the optical vector-matrix multiplier. Since the time to input a number is proportional to its value and at least one of the two non-negative components of each element of the array $\{x_m\}$ will be zero, the throughput rate for bipolar data is

$$\frac{1}{f_o} \sum_{m=1}^{M=10} |x_m|, \text{ where } f_o = 1 \text{ mhz and } -128 \leq x_m \leq 127 \text{ for the laboratory system}$$

fabricated.

The total power incident on the n-th detector element is

$$O_n(t) = \frac{L}{M} \sum_{m=1}^{M=10} t_{mn} b_{mn} I_m(t) \quad (3.13)$$

where $I_m(t)$ is specified by (3.11); b_{mn} are the values of the matrix mask; t_{mn} is the non-uniform spatial transmittance of the fiber optic element and L is the optical efficiency. Since the LEDS are pulsed sequentially in our system to simplify hardware design and fabrication, the photodiodes must time-integrate (3.13) over the entire period T .

The photodetector support electronics [subsystem (ii)] contains a bank of 10 resettable op-amp integrators which sum each of the photocurrents over this time interval. At time $t=T$, the output voltage of the n-th integrator is

$$O_n(T) = \frac{1}{C} \int_0^T [r_n O_n(t) + i_n(t)] dt \quad (3.14)$$

where r_n is the responsivity of the n-th photo diode, $O_n(t)$ is given by (3.13), C is the integration capacitance and $i_n(t)$ is the noise current introduced by the n-th photodiode and support electronics. The variations in fiber optic transmittance t_{mn} , the peak power P_m radiated by the LEDS and the detector responsivity r_n will be compensated for by a static correction mask placed in contact with the matrix mask and by electronic correction to be performed in the microprocessor. Letting C_n be the signal component of (3.14), we obtain for the output

$$C_n = \left(\frac{r L_o}{f_o M C} \right) \sum_{m=1}^{M=10} a_m b_{mn} \quad (3.15)$$

where r is the average detector responsivity and ρ is the average peak power radiated by the LEDS. In practice, the gain $(r L_o / f_o M C)$ is set equal to unity by proper selection of the integration capacitance C and adjustment

of the clock frequency f_0 . The analog voltages $v_n(T) \cong C_n$ are read-out sequentially by the MUX, converted into digital form by the A/D converter in subsystem (iii) and fed back into the microprocessor.

The microprocessor controller [subsystem (iv)], which is depicted in the hardware block diagram of Figure 3.7, is responsible for the scheduling and execution of all operations in the computation cycle of the IOP. These operations include: LED preprocessing and correction, optical vector-matrix multiplication, detector post-processing, bias removal, rescaling and vector addition. The microprocessor contains a custom designed arithmetic unit consisting of a 16 bit, 300 ns TRW multiplier, a 16-bit arithmetic logic unit (ALU) and a 16K random access memory (RAM) with a row-column address structure. We have arranged these units to simplify software control and data paths.

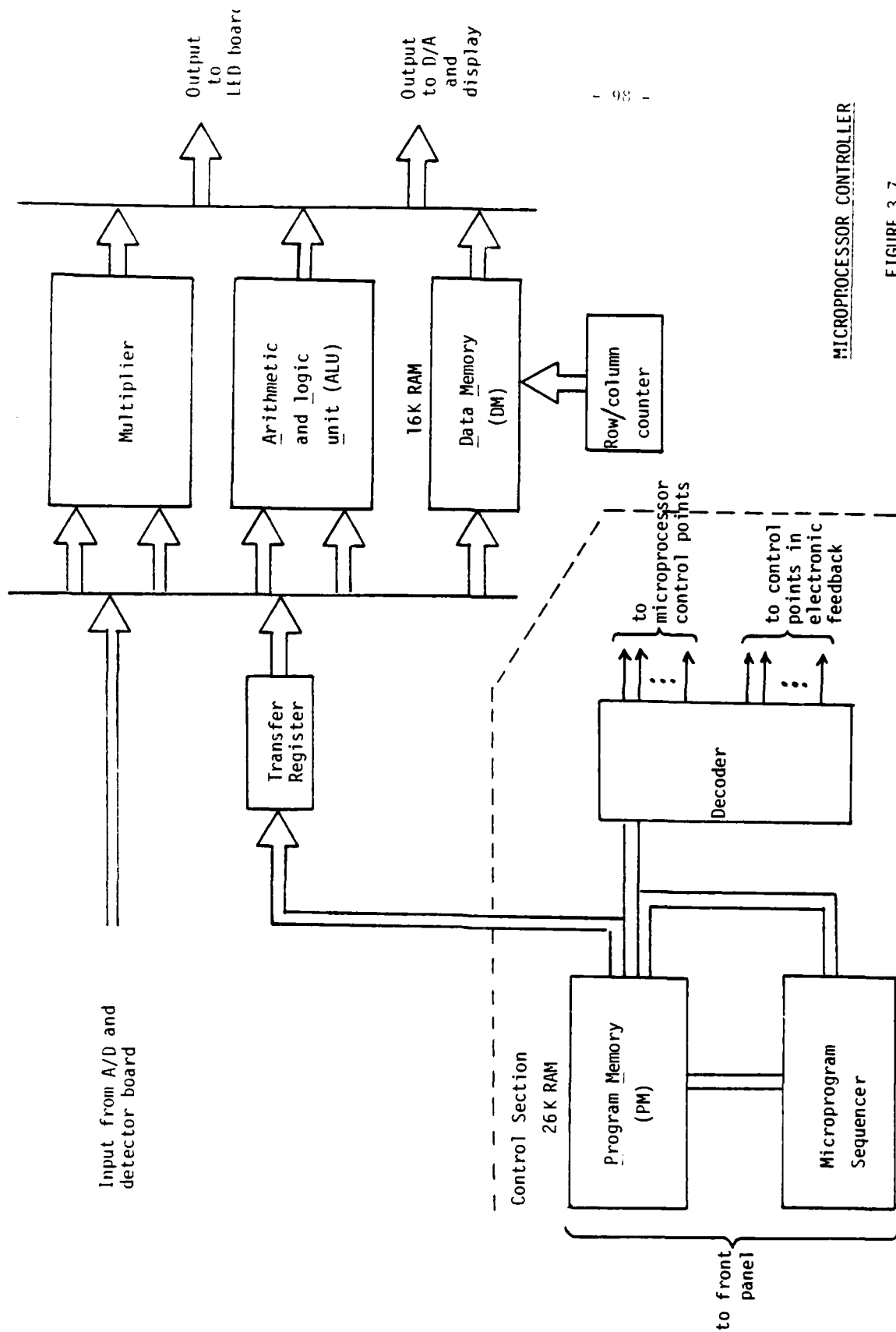
The control section sequences the internal data transfers, arithmetic operations and input/output as well as the operation of the LED and detector boards (i) and (ii) in Figure 3.6. It consists of a 26K RAM to store the microprograms which operate the IOP, a Fairchild 9408 LSI microprogram sequencer to execute the stored programs and a 32 line instruction decoder which activates the various control points in the system. The instructions are executed as either evokes to activate control lines, conditional branches or jumps to a subroutine.

An extensive IOP program selection exists for component and system tests and for system operation. A console (front panel) is provided to load the programs into microprogram RAM and contains all necessary operator controls to start, stop and reset the microprocessor. The microprocessor controller and front panel contain 160 IC's which consume about 50 watts of power. The cycle time for any microinstruction is 300 ns. A photograph of the entire laboratory IOP system is shown in Figure 3.8.

3.3.3 System Operation

Before starting the microprocessor, the vector data \underline{y} and $\underline{x}(0)$ are loaded into the data memory (DM) and the scalar parameters ω , \underline{h} and $(\bar{h}-\underline{h})$ are loaded into program memory (PM) via the front panel. The LED and detector correction data, $\{\frac{1}{p_m}\}$ and $\{\frac{1}{r_n}\}$, are permanently stored in the DM.

The memory maps are shown in Tables 3.2 and 3.3. The data memory (Table 3.2) contains these data as well as the current bipolar iterate $\underline{x}(k)$, its positive \underline{a}_1 and negative \underline{a}_2 vector components, the optically computed \underline{c}_1 and \underline{c}_2 vectors



MICROPROCESSOR CONTROLLER

FIGURE 3.7

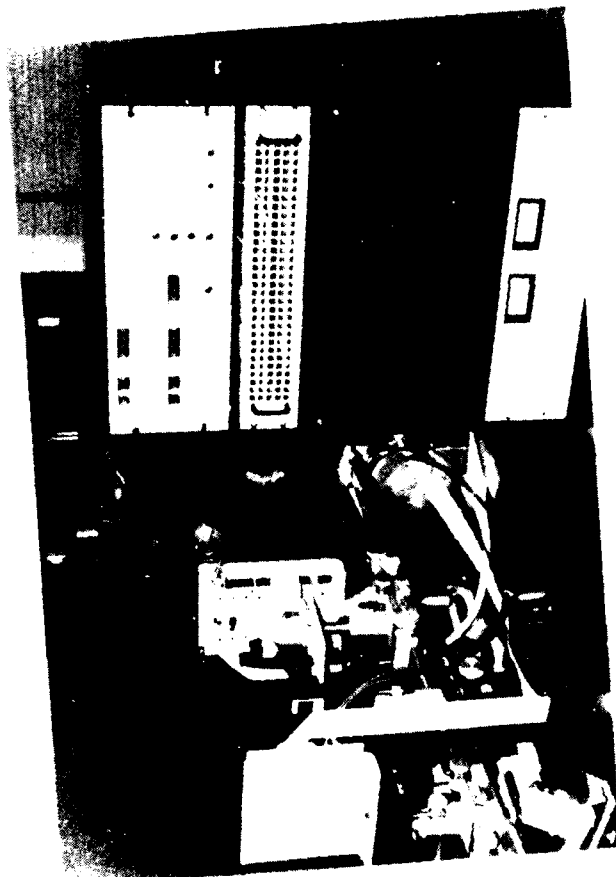


FIGURE 3.8
PHOTOGRAPH OF THE ENTIRE LABORATORY IO¹ SYSTEM

		Column Address 0-9	
Row address	0	Input vector \underline{y}	IOP data
	1	k-th iterate $\underline{x}(k)$	
	2	LED correction $\{1/p_m\}$	
	3	positive optical input \underline{a}_1	
	4	negative optical input \underline{a}_2	
	5	positive optical output \underline{c}_1	
	6	negative optical output \underline{c}_2	
	7	photodiode correction $\{1/r_n\}$	
	8	k-th bipolar output $\underline{y}(k)$	

Table 3.2 Data Memory Map: DM(0 : 63, 0 : 16), 16 bits

Address 0-255	Main program
256-1023	Subroutines: (i) vector accumulation (ii) LED pre-processing (iii) Optical V-M multiplication (iv) detector post-processing

Table 3.3 Program Memory Map: PM(0 : 1023), 26 bits

and the resultant bipolar vector-matrix output $\underline{y}(k) = \underline{H} \underline{x}(k)$.

The program structure of the IOP is divided into four major subroutines:

- (i) Vector accumulation
- (ii) LED pre-processing
- (iii) Optical vector-matrix multiplication and
- (iv) Detector post-processing

The subroutines are stored in program memory (Table 3.2) along with subroutines for data loading, initialization and display and special routines for system tests and calibration. The main program (which is essentially a series of calls to the aforementioned subroutines) is located in the top portion of program memory. A flowchart of the computational cycle of the IOP is shown in Figure 3.9. Each block is one of the four subroutines mentioned above. After starting the microprocessor, various system initializations are performed [including $\underline{y}(0) = \underline{0}$]. The vector accumulation [subroutine (i)] electronically subtracts the input vector \underline{y} from the last bipolar vector-matrix product $\underline{y}(k-1)$, multiplies the difference by the acceleration parameter ω and subtracts the result from the last iterate $\underline{x}(k-1)$ to form the new iterate $\underline{x}(k)$.

This new iterate is separated into the optical vectors \underline{a}_1 and \underline{a}_2 which are then electronically multiplied by the LED correction factors $\{\frac{1}{P_m}\}$ in the LED preprocessing routine (ii). Next, two vector-matrix multiplications are performed optically on these data by subroutine (iii). First, the corrected positive component \underline{a}_1 of \underline{x} is fed to the LEDS; \underline{c}_1 is optically computed and stored. The corrected negative component \underline{a}_2 of \underline{x} is then fed to the LEDS and \underline{c}_2 is optically computed and stored. In the detector post-processing routine (iv), these two vectors \underline{c}_1 and \underline{c}_2 are electronically subtracted and multiplied by the detector correction factors $\{\frac{1}{r_n}\}$. The result is then scaled by the factor $(\bar{h}-h)$ and the bias $\underline{h}[\sum_{m=1}^{M=10} x_m(1)]$

is removed from each of the output components $y_n(k)$ to generate the bipolar output $\underline{y}(k)$.

This procedure is repeated until terminated either manually (by the console control) or by a preset break-point in the microprogram. The sequence of iterates $\underline{x}(0)$, $\underline{x}(1)$, $\underline{x}(2)$...can either be displayed in real-time as they are generated or stored in PM for play back and off-line analysis.

Up to 64 iterates can be stored in the present system.

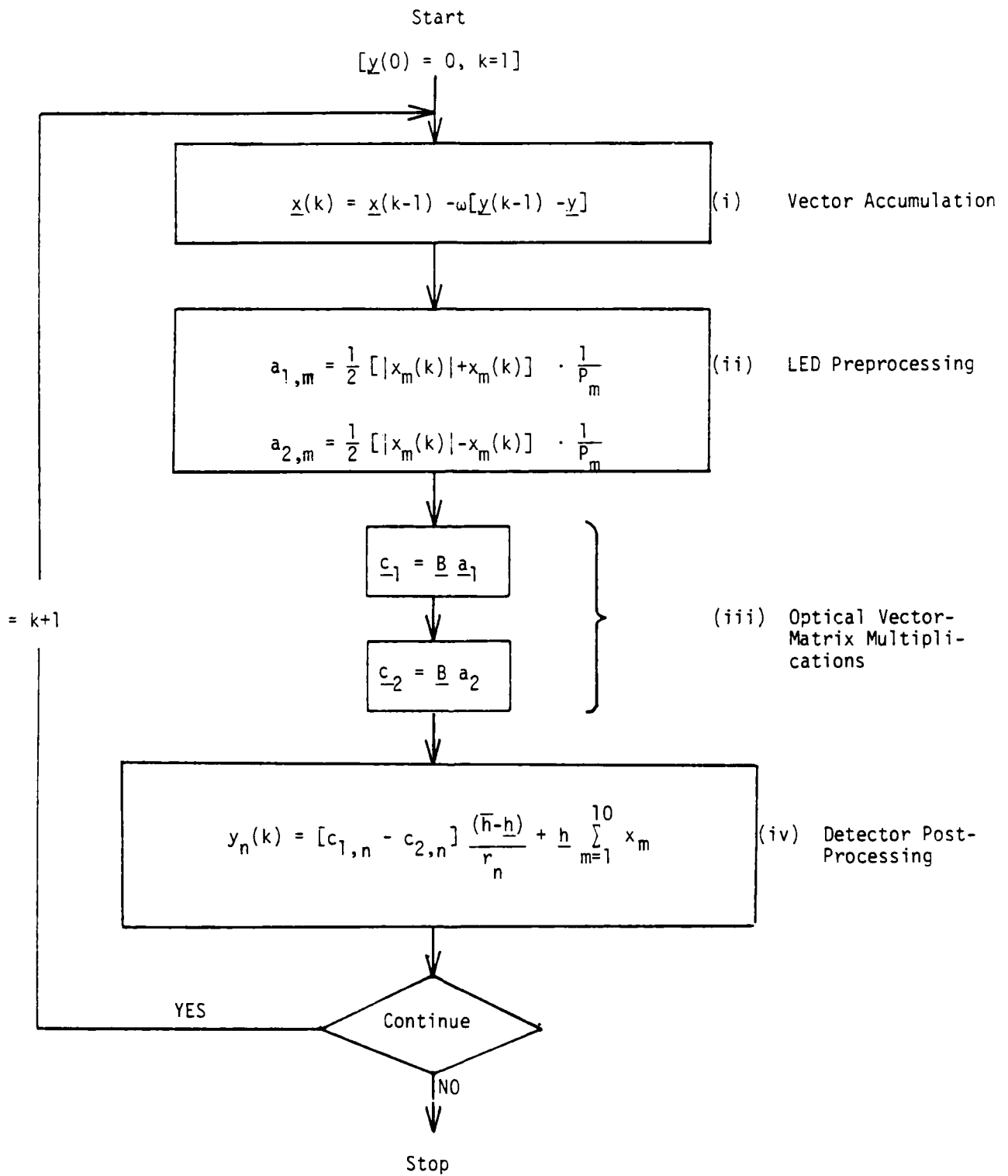


FIGURE 3.9

IOP COMPUTATIONAL CYCLE FLOWGRAPH

3.4 IOP SIMULATOR

Another new feature of this IOP program was to extend its use to include adaptivity in both angle and time. This, combined with the need to handle complex-valued data, increases the required SBWP of the vectors and matrices. We thus used simulation to study such advanced cases. We have written an IOP simulator that is quite interactive. Two levels of simulator, a basic and an advanced version, have been written. Each is described in this section and then several examples of their use are given. Other uses of the simulator are included in Sects. 3.5 and 3.6. All routines are written in Fortran.

The APAR .1. FOR routine (Table 3.4) calculates the covariance matrix \underline{M} , steering vector \underline{s} and the mask transmittance function $\underline{I} - \underline{M}$ from the given input variables. The operator enters the number of antenna elements, the number of vector inputs to be sampled in time and space, the receiver noise power P_r as well as the velocity, angle, position, number M , and strengths P_m of the M noise sources. This defines the set of M noise sources and the receiving array geometry. We also specify the steering vector by the location and angle of the target. From this, we produce the 2-D antenna receiver-processor model of Fig. 3.10. This new model has coordinates (n, n') corresponding to the space and time taps respectively, with N adaptive elements n in space and N' time taps n' for each received element. In Fig. 3.10, the time history flow of the received signal at each element in space is recorded vertically and sampled at N' time intervals. A 5×5 matrix or 2-D space-time antenna grid is used to reduce computations.

We use an incremental space increment (antenna element spacing) of $d = \lambda/2 = 0.5$ and time increments $T = 0.0005$ corresponding to a velocity $v_{\max} = \lambda/4T = \pm 500$.

TABLE 3.4

APAR.1. FOR ROUTINE

```
00100      INTEGER M,MMAX,L,LMAX,N1,N2,L1,L2,NAPS(1:20),MAPT(1:20)
00200      REAL V(0:19),TH(0:19),P(0:19)
00300      REAL PI,T,D,LMDA,N
00400      REAL ARG
00500      COMPLEX CDV(1:20,1:20),STAR(1:20)
00600      PI=3.14159
00700      T=0.001
00800      D=0.5
00900      LMDA=1.0
01000      C      ***THIS SECTION OF CODE INPUTS THE FAR FIELD ANTENNA PATTERN***
01100      C
01200      C
01300      C
01400      C
01500      WRITE(5,5)
01600      5      FORMAT(' ENTER NUMBER OF NOISE SOURCES AND ELEMENTS IN VECTOR ')
01700      READ(5,10) MMAX,LMAX
01800      10     FORMAT(2I)
01900      DO 35 M=1,MMAX
02000      WRITE(5,30) M
02100      30     FORMAT(' ENTER VELOCITY, ANGLE AND STRENGTH OF SOURCE # ',I2)
02200      READ(5,32) V(M),TH(M),P(M)
02300      TH(M)=TH(M)*PI/180.0
02400      32     FORMAT(3F)
02500      35     CONTINUE
02600      40     WRITE(5,45)
02700      45     FORMAT(' ENTER VELOCITY AND ANGLE OF TARGET ')
02800      READ(5,50) V(0),TH(0)
02900      TH(0)=TH(0)*PI/180.0
03000      50     FORMAT(2F)
03100      WRITE(5,70)
03200      70     FORMAT(' TYPE RECEIVER NOISE POWER AND TARGET GAIN ')
03300      READ(5,75) N,P(0)
03400      75     FORMAT(2F)
03500      C      ***WE NOW MAP THE SAMPLED 2D ANTENNA PATTERN INTO A 1D VECTOR***
03600      C
03700      C
03800      C
03900      C
04000      DO 100 L=1,LMAX
04100      WRITE(5,85) L
04200      85     FORMAT(' ENTER SPACE TIME COORDINATE OF ELEMENT # ',I2)
04300      READ(5,90) N1,N2
04400      90     FORMAT(2I)
04500      MAPS(L)=N1
04600      95     MAPT(L)=N2
04700      100    CONTINUE
```

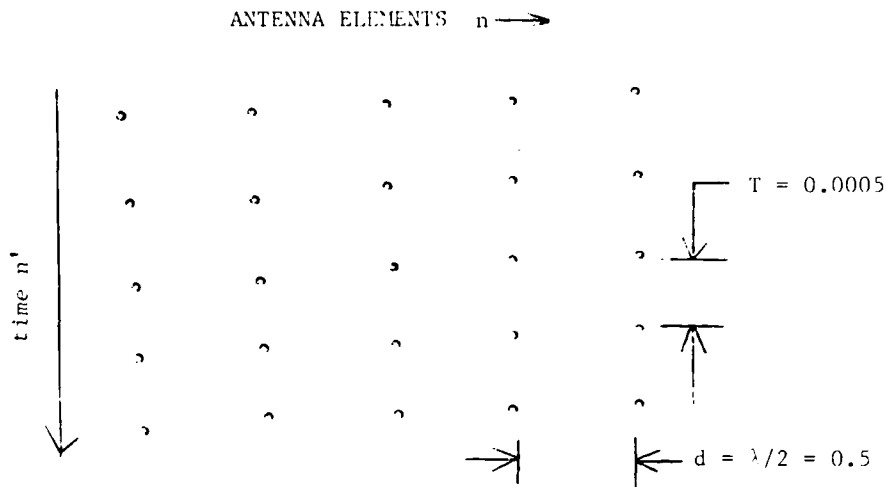


Figure 3.10 2-D space-time antenna model (n, n') .

In terms of conventional units, for a 1 GHz radar, $f = 10^9$ Hz, $\lambda = 0.3$ m, $T = 1.5 \times 10^{-4}$ sec and $v = \pm 500$ m/sec. These units and relationships arise from the relationship for the phase of a signal at element n and time tap n' for a target at angle θ and velocity v and a time delay T between antenna taps given by

$$\phi_{n,n'} = 2\pi(nd\sin\theta + 2vTn')/\lambda. \quad (3.16)$$

When $d = \lambda/2$ and $\lambda = 1$, (3.16) becomes

$$\phi_{n,n'} = \pi \left[n \sin\theta + n' \left(v/v_{\max} \right) \right], \quad (3.17)$$

where v_{\max} is the maximum target velocity given by $v_{\max} = \lambda/4T$.

This (n,n') mapping is thus recorded and a 2-D map of (n,n') samples is produced. The vector \underline{s}^* , matrices $\underline{I} - \underline{M}$ and the (n,n') maps are stored in a disk file. To map this 2-D data onto the 1-D vector-matrix system, we must map (n,n') into a 1-D vector \underline{z} . This routine accomplishes this and stores the associated maps on a disk file. The routine to compute \underline{s} and \underline{M} is given in Table 3.5. In Fig. 3.10, we see that a 25 element 5×5 space-time antenna grid is possible.

TABLE 3.5

ROUTINE TO COMPUTE S & M

```
04800 C      ***COMPUTES STEERING VECTOR AND COVARIANCE MATRIX***
04900 C
05000 C
05100 C
05200 C
05300      DO 150 L=1,LMAX
05400      ARG=2*PI*(MAPT(L)*V(0)*T+MAPS(L)*D*SIN(TH(0)))/LMDA
05500      STAR(L)=CMPLX(SQRT(P(0)),0.0)*
05600      1      CMPLX(COS(ARG),SIN(ARG))
05700 150    CONTINUE
05800      WRITE(5,155)
05900 155    FORMAT(' STEERING VECTOR COMPUTED ')
06000      DO 200 L1=1,LMAX
06100      DO 200 L2=1,LMAX
06200      IF (L1=L2) COV(L1,L2)=CMPLX(N,0.0)
06300      IF (L1#L2) COV(L1,L2)=(0.0,0.0)
06400      DO 200 M=1,MMAX
06500      ARG=2*PI*((MAPT(L2)-MAPT(L1))*V(M)*T+
06600      1      (MAPS(L2)-MAPS(L1))*D*SIN(TH(M)))/LMDA
06700      COV(L1,L2)=COV(L1,L2)+CMPLX(P(M),0.0)*CMPLX(COS(ARG),-SIN(ARG))
06800 200    CONTINUE
06900      DO 202 L1=1,LMAX
07000      DO 202 L2=1,LMAX
07100      COV(L1,L2)=(-1.0,0.0)*COV(L1,L2)
07200      IF (L1=L2) COV(L1,L2)=COV(L1,L2)+(1.0,0.0)
07300 202    CONTINUE
07400      WRITE(5,205)
07500 205    FORMAT(' COVARIANCE MATRIX COMPUTED ')
07600      OPEN(UNIT=20,FILE=' STEER. DAT ')
07700 300    FORMAT(2F)
07800      DO 310 L=1,LMAX
07900      WRITE(20,300) STAR(L)
08000 310    CONTINUE
08100      OPEN(UNIT=21,FILE=' COVAR. DAT ')
08200      DO 320 L1=1,LMAX
08300      DO 320 L2=1,LMAX
08400      WRITE(21,300) COV(L1,L2)
08500 320    CONTINUE
08600      OPEN(UNIT=22,FILE=' ANTMAP. DAT ')
08700      DO 350 L1=1,LMAX
08800      WRITE(22,330) MAPS(L1),MAPT(L1)
08900 330    FORMAT(2I)
09000 350    CONTINUE
09100      STOP
09200      END
```

With 10 LEDs, we can represent 10 steering vectors and 10 adaptive weights. Since complex-valued data is necessary, each element requires four bipolar entries, the plus and minus real and imaginary parts of each vector or matrix element. With the time-sequential bipolar data technique (Sect. 3.2), only two values are necessary per cycle per element. Thus, for each scenario, we can use 5 of the 25 possible samples in the 2-D (n, n') matrix antenna model of Fig. 3.10.

From the APAR .1. FOR routine, we obtain a 2-D to 1-D mapping of (n, n') into i , with any 5 of the 25 possible 2-D sampled matrix positions in Fig. 3.10 possible for entry into the IOP. The third routine in the basic simulator is ITER.1. FOR (Table 3.6). This implements the vector-matrix iterative equation

$$\underline{w}_{i+1} = (\underline{I} - \underline{M}) \underline{w}_i + \underline{s}^* \quad (3.18)$$

in the form

$$\underline{w}_{i+1} = \underline{w}_i + \alpha \left(\underline{M} \underline{w}_i - \underline{s}^* \right), \quad (3.19)$$

where in (3.19), $\underline{M} \underline{w}_i$ is the new vector-matrix product, $\underline{M} \underline{w}_i - \underline{s}^*$ is the difference that is to be reduced to zero and α is the acceleration factor. We discuss selection of α in Sect. 3.5 and its choice in speeding convergence of the algorithm and in forcing the algorithm to converge. In general, if the eigen-values are small, α is chosen large, since this increases the eigen-values and makes convergence faster. Conversely, if the eigen-values are large, i.e. $\alpha > 1$, α is chosen small to decrease all eigen-values to be less than 1 thereby insuring convergence of the iterative algorithm. If one noise source is much larger or smaller than another, large and small eigen-values will result and we must thus scale the matrix \underline{M} such that the largest eigen-value fits within the dynamic range of \underline{M} and such that all eigen-values are less than 1. When the dynamic range of the eigen-values is large, a loss in system noise

TABLE 3.6

ITER.1. FOR ROUTINE

```
00100  C      ***THIS PROGRAM SIMULATES THE IDEAL APAR ITERATIVE
00200  C      PROCESSOR AND EITHER PRINTS THE WEIGHTS AT EACH ITERATION
00300  C      OR DISPLAYS AN ISOMETRIC PLOT OF THE FAR FIELD PATTERN. **
00400  C
00500  C
00600      INTEGER L1, L2, LMAX, INCRE, ITER, K1, K2, KMAX
00700      INTEGER TEST, DISP, MAPS(1: 20), MAPT(1: 20)
00800      REAL PI, T, D, LMDA
00900      REAL ARG
01000      REAL F(0: 64, 0: 64)
01100      REAL RATIO1, RATIO2
01200      COMPLEX STAR(1: 20), COV(1: 20, 1: 20), WOLD(1: 20), WNEW(1: 20)
01300      COMPLEX E(0: 64, 0: 64)
01400      PI=3.14159
01500      T=0.001
01600      D=0.5
01700      LMDA=1.0
01800  C
01900  C
02000  C
02100      WRITE(5, 1)
02200  1      FORMAT(' ENTER LENGTH OF VECTOR AND THE DISPLAY INCREMENT ')
02300      READ(5, 2) LMAX, INCRE
02400  2      FORMAT(2I)
02500      OPEN(UNIT=20, FILE=' STEER. DAT ')
02600  3      FORMAT(2F)
02700      DO 4 L1=1, LMAX
02800          READ(20, 3) STAR(L1)
02900  4      CONTINUE
03000      OPEN(UNIT=21, FILE=' COVAR. DAT ')
03100      DO 5 L1=1, LMAX
03200          DO 5 L2=1, LMAX
03300              READ(21, 3) COV(L1, L2)
03400  5      CONTINUE
03500      WRITE(5, 10)
03600  10      FORMAT(' TYPE 1 TO PRINT W AND 0 TO PLOT E ')
03700      READ(5, 12) DISP
03800  12      FORMAT(I1)
03900      ITER=0
04000      OPEN(UNIT=22, FILE=' ANTMAP. DAT ')
04100      DO 14 L1=1, LMAX
04200          READ(22, 13) MAPS(L1), MAPT(L1)
04300  13      FORMAT(2I)
04400  14      CONTINUE
```

adaptivity may result by the above reduction process to accommodate the dynamic range of the M mask. This issue will be addressed in later phases of this work.

The routine ITER.1. FOR, reads the disk data from the output of APAR.1. FOR, calculates the \underline{w}_i and also $E(\cdot)$ at each iteration due to the present \underline{w}_i estimate. The latter operation is performed by the routine in Table 3.7.

Many sample noise scenarios were run using this simulator. In all cases, the isometric outputs of the $E(\cdot)$ adaptive antenna pattern are shown after different numbers of iterative cycles. For the different cases to be described here, the output $E(\cdot)$ antenna pattern that results from the applied adaptive weights are shown after 100 iteration cycles. We first consider the case of adaptivity in θ only. We assume one noise source at 12^0 and the signal or steering vector at 45^0 with receiver noise power $P_r = 0.1$ and noise power $P_1 = 0.4$. The results are shown in Fig. 3.11 after 100 iterations. The pattern shows a null at 12^0 (the angle of the noise source) as expected. In Fig. 3.12, we show similar output results for the case of four noise sources at $+30^0$ and $+60^0$ each with power $P = 0.1$ for the case of a boresight steering vector after 100 iterations. Again good results are observed using the iterative algorithm. Nulls appear at the four correct noise source angles in the adaptive pattern shown.

We next considered the case of three noise sources at $v = 250$, $\theta = -45^0$; $v = 250$, $\theta = +45^0$; and $v = -250$, $\theta = +45^0$. For these cases of a pattern with noise sources at different angles and frequencies, 2-D adaptivity is required. After computing the associated M and the resultant adaptive weights, the resultant adaptive antenna pattern $E(\cdot)$ shown in Fig. 3.13 was obtained. This pattern shows nulls at the correct velocity and angle coordinates corresponding to the three noise sources noted above. It is thus in excellent agreement with the results predicted by theory.

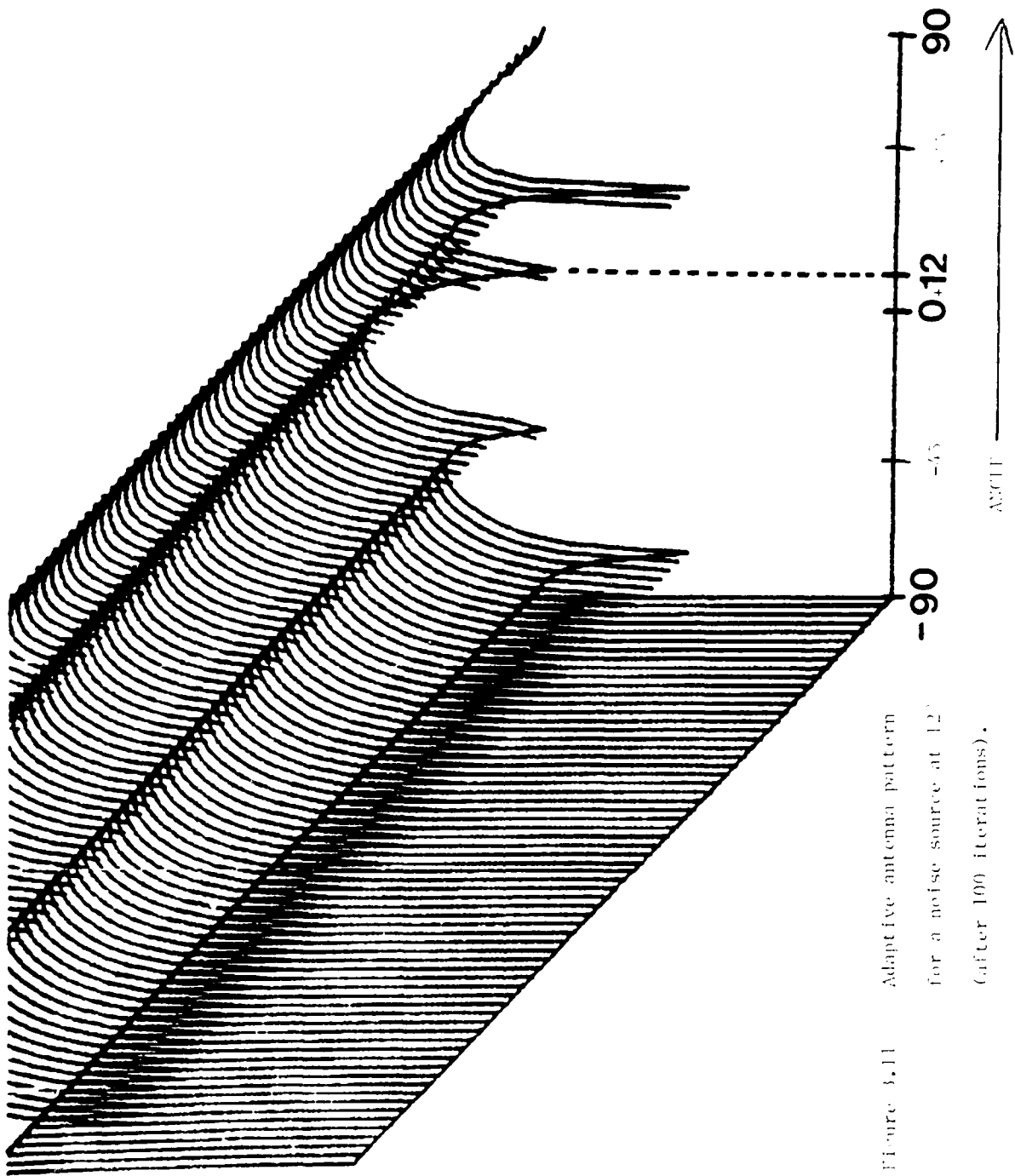
TABLE 3.7

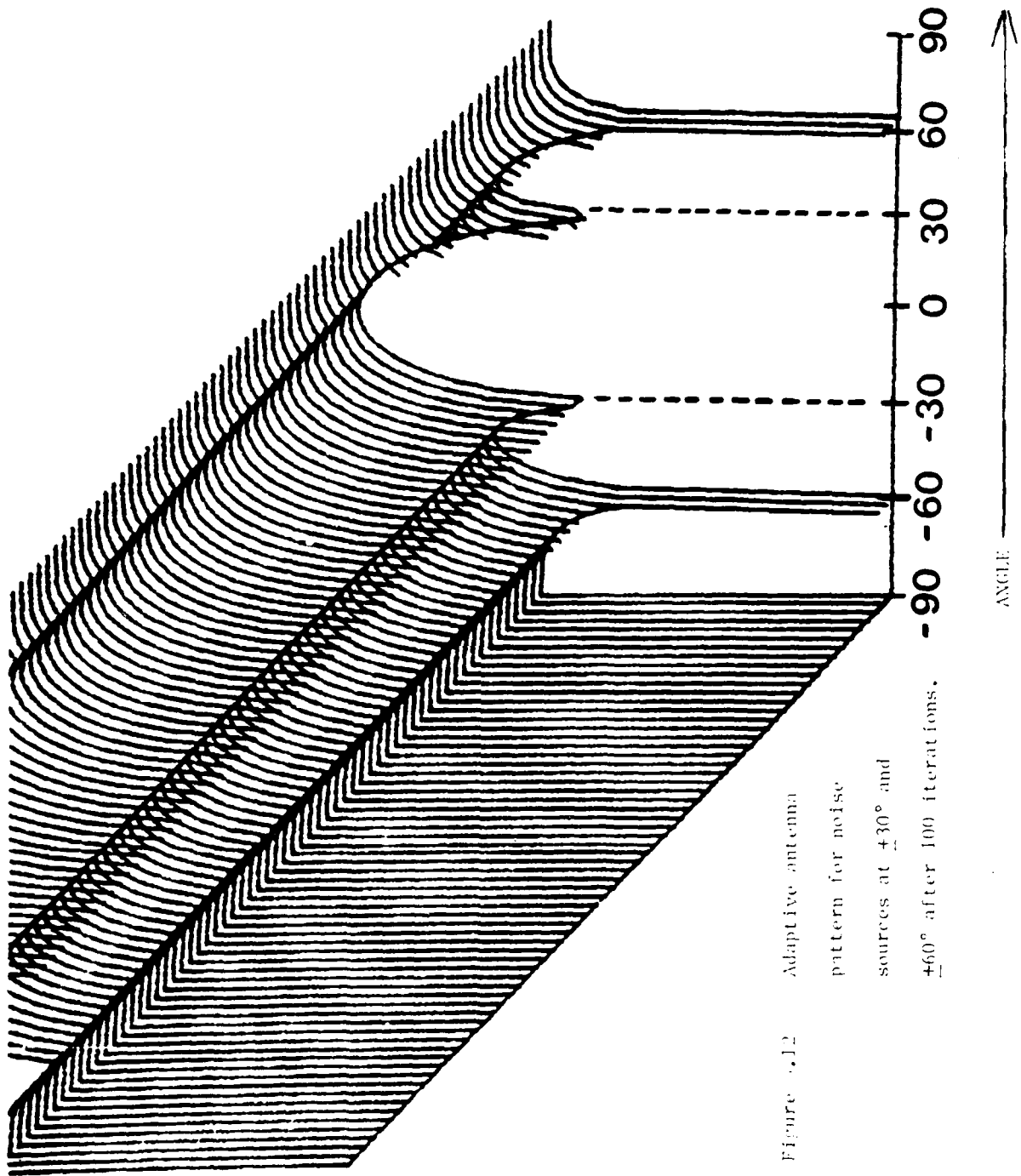
ROUTINE TO COMPUTE E(4)

```

04500 C      ***THIS BLOCK COMPUTES THE ADAPTIVE ANTENNA WEIGHTS**
04600 C
04700 C
04800 C
04900 C
05000      DO 15 L1=1,LMAX
05100      WOLD(L1)=(0.0,0.0)
05200 15      CONTINUE
05300 16      DO 50 L1=1,LMAX
05400      WNEW(L1)=STAR(L1)
05500      DO 50 L2=1,LMAX
05600      WNEW(L1)=WNEW(L1)+COV(L1,L2)*WOLD(L2)
05700 50      CONTINUE
05800      IF (MOD(ITER,INCRE)#0) GO TO 87
05900      IF (DISP==0) GO TO 62
06000      WRITE(5,52) ITER
06100 52      FORMAT(' ITERATION # ',I)
06200      DO 60 L1=1,LMAX
06300      WRITE(5,55) WNEW(L1)
06400 55      FORMAT(2F)
06500 60      CONTINUE
06600      GO TO 79
06700 C      ***THE FOLLOWING SECTION CALCULATES THE FAR FIELD
06800 C      ANTENNA PATTERN AND PRODUCES AN ISOMETRIC PLOT***
06900 C
07000 C
07100 C
07200 C
07300 62      KMAX=64
07400      DO 72 K1=0,KMAX
07500      DO 72 K2=0,KMAX
07600      E(K1,K2)=(0.0,0.0)
07700      DO 70 L1=1,LMAX
07800      RATIO1=FLOAT(K1)/FLOAT(KMAX)
07900      RATIO2=FLOAT(K2)/FLOAT(KMAX)
08000      ARG=2*PI*(MAPS(L1)*D*SIN(PI*(RATIO1-0.5))+
08100      1      MAPT(L1)*(RATIO2-0.5))/LMDA
08200      E(K1,K2)=E(K1,K2)+WNEW(L1)*CMPLX(COS(ARG),-SIN(ARG))
08300 70      CONTINUE
08400      F(K1,K2)=REAL(E(K1,K2)*CONJG(E(K1,K2)))
08500      F(K1,K2)=5+ALOG10(F(K1,K2))
08600 72      CONTINUE
08700      CALL PLOT2D(64,64,F)
08800 79      WRITE(5,80)
08900 80      FORMAT(' TYPE 1 TO CONTINUE ITERATION')
09000      READ(5,85) TEST
09100 85      FORMAT(I)
09200      IF (TEST==0) STOP
09300 87      ITER=ITER+1
09400      DO 90 L1=1,LMAX
09500      WOLD(L1)=WNEW(L1)
09600 90      CONTINUE
09700      GO TO 16
09800      END

```





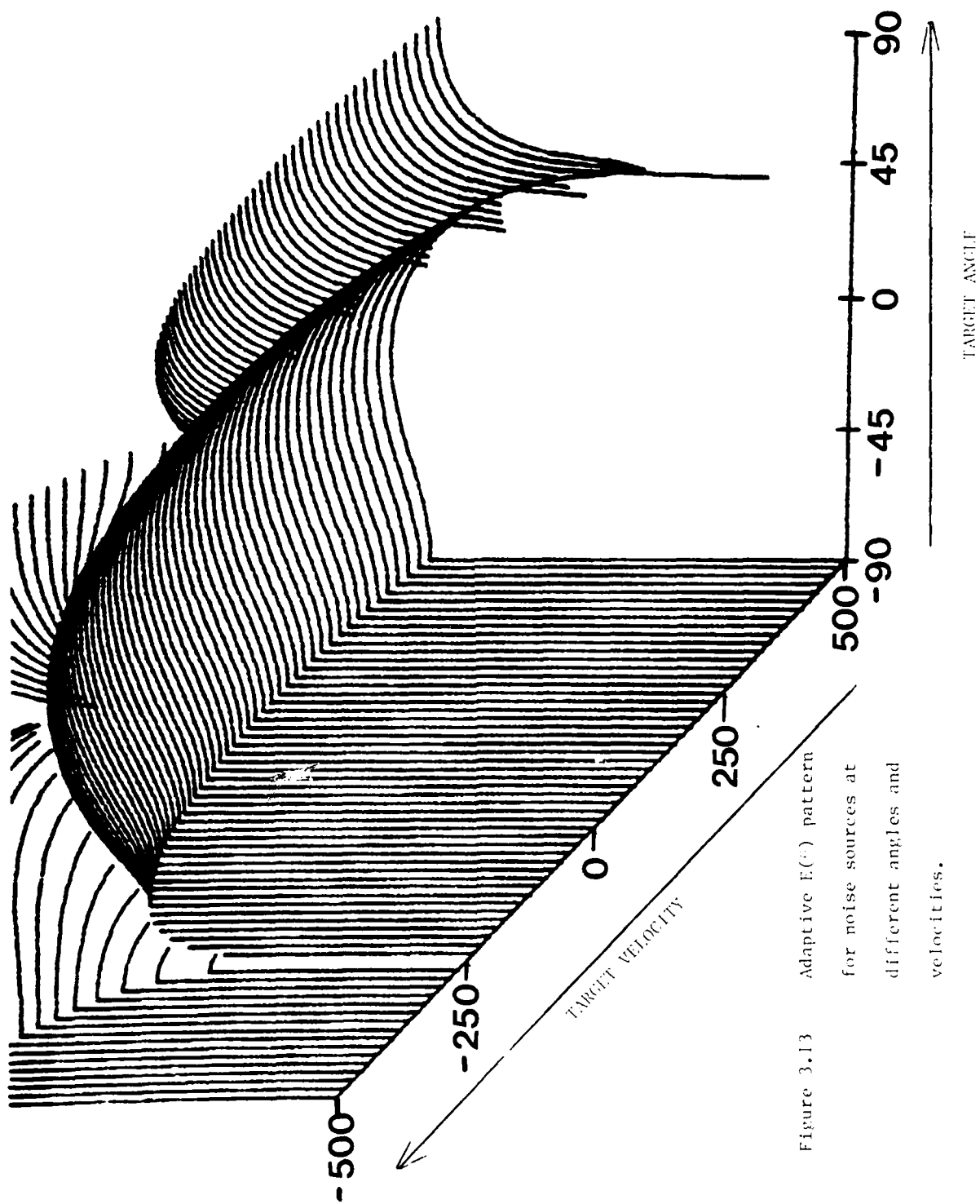


Figure 3.13 Adaptive $E(\cdot)$ pattern for noise sources at different angles and velocities.

This simulator is thus a general routine with increased flexibility and increased real-time user interaction that includes additive receiver noise, the ability to compute M , $I - M$, the iterative weights \underline{w}_i at each iteration and the resultant $E(\cdot)$ adaptive far-field noise pattern that results from the given \underline{w}_i and \underline{g} . An advanced version of this simulator includes the effects of additive and multiplicative noise error sources in the hardware IOP, computes the additive weights and $E(\cdot)$ with and without these error sources, the processing gain or output SNR of the antenna pattern and hence the effect of IOP error sources on the performance of the APAR system. We describe a demonstration of this advanced simulator in Sect. 3.6 to analyze the accuracy and performance of the IOP laboratory demonstration experiments.

3.5 SYSTEM OPERATION

In this section, we consider several different aspects of the IOP system and their effect on the system's design and its performance. In Sect. 3.5.1, we describe the setup of the IOP, measurements of its accuracy and the corrections employed in the experimental system assembled. In Sect. 3.5.2, we discuss convergence of the IOP, selection of the acceleration factor α to improve the rate of convergence and how to determine α from the eigen-values of M . Simulation results are then included to verify these issues.

3.5.1 System Calibration

Upon receiving the different system components and assembling the IOP, we performed many component and system checks and calibrations. In this sub-section, we highlight the major calibration tests and the eventual corrections used. In each case, special micro-processor sub-routines were written to provide the individual required system and component tests noted.

We first measured and corrected for the LED non-uniform saturation levels. A test was made by pulsing each LED on in sequence for $256 \times 4 = 1$ msec (the PWM width corresponding to the largest input value). With no P_2 mask present, the ten fiber optic outputs in each row were measured in sequence with the detector butted against the mask. An array of $10 \times 10 = 100$ voltage values was then obtained. The current into each LED was then adjusted such that each LED produced the same 10 photo detector voltage output readings of 5 volts corresponding to one half the saturation level of the output from the detector board. This ensured that the system was operating in its linear range (one half of saturation voltage). These LED corrections were then entered into a RAM in the micro-processor electronic feedback system and used for all future corrections.

After these corrections were made, the residual system errors were measured. For the 100 output photo detector readings after RAM correction of the LEDs, we measured a standard deviation in the data of 0.093 with a mean value of 6.065. This corresponds to a non-uniformity of 0.8%. In our advanced simulator, we included this as a multiplicative error in the mask transmittance. This measurement and correction technique corrects for the differences in LED saturation levels and the differences in transmittance of the different fiber optic rows. This $10 \times 10 = 100$ element error matrix with 0.8% non-uniformity could be corrected for by a fixed mask in contact with the B matrix at P_2 . However, computer plotters and other optical film recorder systems to which we presently have access cannot correct for such an accuracy of less than 1% in amplitude. In the latter phase of this project, we will use our film recorder to produce a correction mask to the necessary accuracy. Many system corrections are necessary to the film recorder before this performance is possible however.

We then measured the total output detector noise. This measurement was performed with no light incident on the photo detector and with coaxial cable connected

to the detector board outputs. We measured a 10 mV peak-peak variance. This represents the temporal noise variance on a fixed noise pattern. The latter noise is removed by the 2-cycle sequential subtraction technique used. The former temporal noise represents an 0.16% error source. We entered this noise error source as additive noise into our advanced simulator to determine its effect on the computed weights w_i and the resultant $E(\cdot)$ adaptive antenna pattern. To do this, we produce a random vector with a peak-peak noise of 10 mV and add this to the vector-matrix output. This low measured value of temporal detector noise is so small that no additive noise distribution (Gaussian, etc.) noise modelling and analysis was performed. The peak detector voltage is 6V and its LDR is 625:1.

We then measured the linearity of the system by increasing the LED pulse width from 0-500 μ sec (with 100 mA drive current used) and measure the photo detector output. The linearity of the system was essentially perfect as expected because of the excellent linearity of the external integrator used.

A system light level budget analysis was then performed to determine the LED power necessary. We denote the output detector voltage by

$$V = RP\tau t/c, \quad (3.20)$$

where R is the responsivity of the detector (0.5 A/W), P is the input power from one LED, $\tau t_{\max} = 4 \times 128 = 512$ μ sec is the maximum input pulse duration, and C = 0.75 pF is the external capacitor used in the integrator. For V = 5v (saturation of the photo detector), we find $P = 1.5 \times 10^{-6}$ W of power from one LED to be necessary (one LED on is the case to be considered for saturation of the output detector for normalization). The LED used provides 1.5×10^{-3} W of intensity. However, a large (approximately a factor of 10) loss occurs in coupling the LED outputs to the fiber optic connector. This is due to the 0.1" spacing used between the LEDs and the fiber optic connector. This spacing was chosen to improve the uniformity of the light into each

fiber optic bundle. An additional factor of 10 loss occurs when the LED drive currents are adjusted to produce equal outputs. The quartz window on the detector and other losses add to reduce the light reaching the photo detector. We then adjusted the pulse width such that a 5 volt saturated output voltage was obtained from the photo detector when $T = 512 \mu \text{ sec}$ was used. The above values satisfy these power requirements and sensitivity values for the system and were thus used in the final design.

These additive 0.16% and multiplicative 0.8% residual errors left after RAM correction were then used in the advanced simulator and a mean square error in the adaptive weights of 0.1% was computed. This small error in the \underline{w}_i due to the residual additive and multiplicative noise error sources is so low that it is expected to produce no significant effect. We discuss this issue further in Sect. 3.6. We note for now that the range or spread in the eigen-values of \underline{M} can effect this, that additive noise (detector) can effect the maximum eigen-value, and that multiplicative noise (mask) can cause stability problems and oscillations.

3.5.2 IOP Convergence

In formulating an IOP solution for the space and time adaptive APAR problem, we write the L receiver inputs (complex-valued) as

$$\underline{x} = \begin{bmatrix} x_1, \dots, x_L \end{bmatrix}^T, \quad (3.21)$$

where the L choices can be any of the $N_s N_t$ space-time taps in the 2-D antenna array model. The covariance matrix

$$\underline{M} = E \begin{bmatrix} \underline{x}^* & \underline{x}^T \\ \underline{x} & \underline{x}^T \end{bmatrix} \quad (3.22)$$

is then computed. We then estimate the largest eigen-value of M from

$$\lambda_{\max} = \left[\sum_{i=1}^L \sum_{j=1}^L |m_{ij}|^2 \right]^{1/2}. \quad (3.23)$$

The IOP then computes

$$\begin{aligned} \underline{w}_{k+1} &= \underline{w}_k + \epsilon (\underline{s}^* - \underline{M} \underline{w}_k) \\ &= (\underline{I} - \epsilon \underline{M}) \underline{w}_k + \epsilon \underline{s}^*. \end{aligned} \quad (3.24)$$

For (3.24) to converge to the solution

$$\underline{w} = \underline{M}^{-1} \underline{s}^*, \quad (3.25)$$

we require the eigen-values λ_i of M to satisfy

$$0 < |1 - \epsilon \lambda_i| < 1 \quad (3.26)$$

for $i = 1, \dots, L$. For maximum monotonic convergence at the fastest possible rate, we choose

$$\epsilon = 1/\lambda_{\max} \quad (3.27)$$

and we approximate the number of iterations k necessary by

$$k_{\max} = 1/\log(1 - P_r), \quad (3.28)$$

where P_r is the receiver noise.

To prove (3.27) and (3.28), we consider a two element antenna with receiver noise power P_r and a single wide-band jammer at ω_1 with power P_1 . For this case

$$\underline{M} = \begin{pmatrix} P_r + P_1 & P_1 \exp(+j\gamma) \\ P_1 \exp(-j\gamma) & P_r + P_1 \end{pmatrix}, \quad (3.29)$$

where $\gamma = (2\pi d/\lambda) \sin \theta$. The eigen-values λ of \underline{M} are the solution of

$$\det[\underline{M} - \underline{I}\lambda] = 0 \quad (3.30)$$

for the case in (3.29), the solution of (3.30) yields

$$\lambda_1 = P_r + P_1 + \sqrt{(P_r + P_1)^2 - P_r^2 - 2P_r P_1} \quad (3.31a)$$

$$\lambda_2 = P_r + P_1 - \sqrt{(P_r + P_1)^2 - P_r^2 - 2P_r P_1}. \quad (3.31b)$$

If $P_r \gg P_1$,

$$\lambda_1 \approx 2P_1 + P_r \quad (3.32a)$$

$$\lambda_2 \approx P_r. \quad (3.32b)$$

If $P_r \ll P_1$,

$$\lambda_1 = \lambda_2 = P_r + P_1. \quad (3.33)$$

In the usual case, $P_r \ll P_1$, and (3.26) and (3.31) yield

$$0 < \gamma(2P_1 + P_r) < 2, \quad (3.34)$$

or approximately

$$0 < 2\gamma P_1 < 2, \quad (3.35)$$

or for $\gamma = 1$ (no scaling)

$$P_1 < 1. \quad (3.36)$$

Continuing the proof of (3.27) and (3.28), we next determine λ_{\max} for this case. Since \underline{M} is Hermetian, λ_{\max} is given by (3.23),

$$\lambda_{\max} = 2P_1 + P_r = \lambda_1 \quad (3.37)$$

for $P_r = P_1$. For the case of one jammer of power P_1 and L adaptive weights each with noise P_r ,

$$\lambda_{\max} = LP_1 + P_r. \quad (3.38)$$

We next consider the k -th iterative estimate of the weights \underline{w}_k and compare it to the final steady state solution \underline{w} to determine the choice of ϵ and its relation to λ . For a boresight ($\phi = 0$) target direction, $\underline{s}^* = [1, 1]^T$ and $\underline{a}\underline{s}^* = [\alpha_1, \alpha_2]^T$ and we find

$$\underline{w}_k = \begin{bmatrix} \frac{P_1(1 - e^{-j\gamma}) + P_r}{2(P_1P_r + P_r^2)} - \frac{(1 - e^{-j\gamma})(1 - \alpha_1)^{k-1}}{2\lambda_1} - \frac{(1 - e^{-j\gamma})(1 - \alpha_2)^{k-1}}{2\lambda_2} \\ \frac{P_1(1 - e^{+j\gamma}) + P_r}{2(P_1P_r + P_r^2)} - \frac{(1 - e^{+j\gamma})(1 - \alpha_1)^{k-1}}{2\lambda_1} - \frac{(1 - e^{+j\gamma})(1 - \alpha_2)^{k-1}}{2\lambda_2} \end{bmatrix} \quad (3.39)$$

Substituting (3.32), we find that as k approaches infinity, the terms in k approach zero and in steady state

$$\underline{w} = \begin{bmatrix} P_1(1 - e^{-j\gamma})/2P_1P_r \\ P_1(1 - e^{+j\gamma})/2P_1P_r \end{bmatrix}. \quad (3.40)$$

Comparing (3.39) and (3.40), we see that for a rapid convergence we can set the acceleration factor in (3.24)

$$\epsilon = 1/\lambda_{\max} = 1/\lambda_1, \quad (3.41)$$

as in (3.27). Then, the terms in $(1 - \alpha_1)^{k-1}$ in (3.39) equal zero. The remaining

transient terms in (3.39) become

$$\left[1 - \exp(\pm j\theta_1)\right] \left[1 - \lambda_2/\lambda_1\right]^{k-1} / 2P_r \quad (3.42)$$

and are thus decreased in magnitude (after k iterations) by the factor

$$\text{or} \quad \frac{\left[\lambda_1(1 - \lambda_2)/(1 - \lambda_2)\right]^k}{\left[(1 - \lambda_2)/(1 - \lambda_2/\lambda_1)\right]^k} \quad (3.43)$$

As λ_1 approaches λ_2 , the choice in (3.41) most improves the rate of convergence of \underline{w}_k to the steady state solution in (3.40).

Having derived (3.27), we now consider the proof of (3.28). We first find the performance measure of our adaptive array as the SNR, the ratio of the power in the direction of the target of strength P_0 to the sum of the noise power in the antenna due to the M noise sources of strengths P_m at angles θ_m . Thus, at iteration k of the IOP,

$$\text{SNR}(k) = \frac{P_0}{\sum_m P_m} \frac{|E_k(\theta)|^2}{|E_k(\theta_m)|^2} \quad (3.44)$$

In (3.44), we denote the antenna pattern at iteration k by $E_k(\theta)$. This is easily found from the weights \underline{w}_k at iteration k by a simple Fourier transform as

$$E_k(\theta) = \sum_{n=1}^L w_k(n) \exp \left[-j2\pi d \sin(\theta) n \right] \quad (3.45)$$

In our performance measure of the use of the IOP system, specifically the iteration technique, we define the process gain PG of the system as the improvement obtained in the antenna SNR as a function of the number of iterations. We thus define PG(k) as a ratio of the SNR at iteration k to the initial SNR at iteration zero, i.e.

$$\text{PG}(k) = \frac{\text{SNR}(k)}{\text{SNR}(0)} \quad (3.46)$$

We now consider the proof of (3.28) by considering the effect of the choice of α on PG. We denote the number of iterations necessary to achieve the given PG without scaling by k_N and the number of iterations necessary to achieve the same PG with scaling by k_S . Then, from (3.43), we can relate k_S to k_N by

$$k_S = \frac{k_N \log \left(\frac{1 - \lambda_2}{1 - \lambda_2/\lambda_1} \right)}{\log \left(\frac{1 - \lambda_2}{1 - \lambda_2/\lambda_1} \right)}, \quad (3.47)$$

where λ_1 and λ_2 are the largest and smallest eigen-values of M . λ_1 and λ_2 are less than 1; from (3.32), $\lambda_2 = P_r$ for $\lambda_1 \gg \lambda_2$; and from (3.27), $\lambda_1 = 1/\alpha$. This shows that

$$\frac{k_S}{k_N} = \frac{1}{\log \left(\frac{1 - P_r}{1 - P_r/\alpha} \right)}. \quad (3.48)$$

Thus, choosing $\alpha = 1/\lambda_1$ will cause a shift in the PG versus k graph. This also verifies (3.28) and shows the proper choice of α is most important when $P_r = 1$ or $\lambda_{\min} = \lambda_{\max}$. In other cases, choosing $\alpha = 1/\lambda_1$ speeds up convergence by decreasing the number of iterations necessary by the factor

$$\frac{\log \left(\frac{1 - \lambda_{\min}/\lambda_{\max}}{1 - \lambda_{\min}} \right)}{\log \left(\frac{1 - \lambda_{\min}}{1 - \lambda_{\min}} \right)}. \quad (3.49)$$

We next investigated the above issues using the simulator (Sect. 3.4). We first considered the effect of scaling on PF(k) for the case of one jammer at 10^0 . We assumed receiver noise $P_r = 0.001$, signal power $P_0 = 0.001$ and different noise interference powers $P_1 = 0.1, 0.01$ and 0.001 . We calculated PG(k) with and without scaling versus iteration number k . For these scaling cases, we used $\alpha = 1/\lambda_1$ and for the other cases $\alpha = 1$. In these cases, λ_{\max} was less than 1 and scaling by α only affected the convergence rate rather than whether the IOP would converge to the steady state

solution. In all cases, the use of scaling improved the number of iterations necessary for the system to converge to steady state. In Fig. 3.14, we show one example of the outputs obtained. In all cases, PG versus k with scaling produced a shifted version of the PG versus k curve with no scaling as predicted by (3.47) and (3.48). In Fig. 3.14, $P_1 = 0.05$ and $P_2 = 0.001$ and we find $k_S = 0.05 k_n$ in (3.47). Fig. 3.14 shows that in this case, the use of scaling improves the number of iterations necessary for convergence to steady state by a factor of 20 or $1/0.05$ in perfect agreement with theory. In the case of $P_1 = 0.001$, an even larger speed up factor of 125 was obtained. The null depths obtained in the three cases were 48, 20 and 11 dB for the cases of decreasing jammer noise $P_1 = 0.1$, 0.01 and 0.001.

In general, we expect $P_m \rightarrow P_0$. Also, in many cases, we must choose α to ensure that the IOP converges to steady state. This case arises when $\alpha_{\max} > 1$. From (3.39), for the case of five adaptive elements, as in our antenna model, α_{\max} will be greater than 1 if $P_1 > (1 - P_r)/5$. To address this case of $\alpha > 1$ (here α is a relaxation factor rather than an acceleration parameter as in the case when $\alpha < 1$), we chose $P_r = 0.01 = P_0$ and $P_1 = 0.1, 1.0$ and $10.0 \gg P_0$. The resultant PG versus k graphs are shown in Fig. 3.15. These data show that the k necessary to reach steady state increases as P_1 increases. Likewise, the null depth obtained also increases with k . These results are as expected. In this case, the $(1 - \alpha P_r)$ eigenvalues of the system increase due to the scaling and the $(1 - \alpha P_r)^k$ terms in \underline{M} decrease less rapidly. Other analogous cases were simulated and showed the same trends that increasing P_m is similar to decreasing P_r since both increase k and $PG(k)$.

We also investigated the PG or null depth possible for different separations between the target and signal for the case $P_0 = 0.01$ and $P_1 = 1.0$ with $P_r = 0.01$. Scaling was used and the results are shown in Fig. 3.16 for the case of $k = 1000$ and different angular separations θ between the target (beam steering direction)

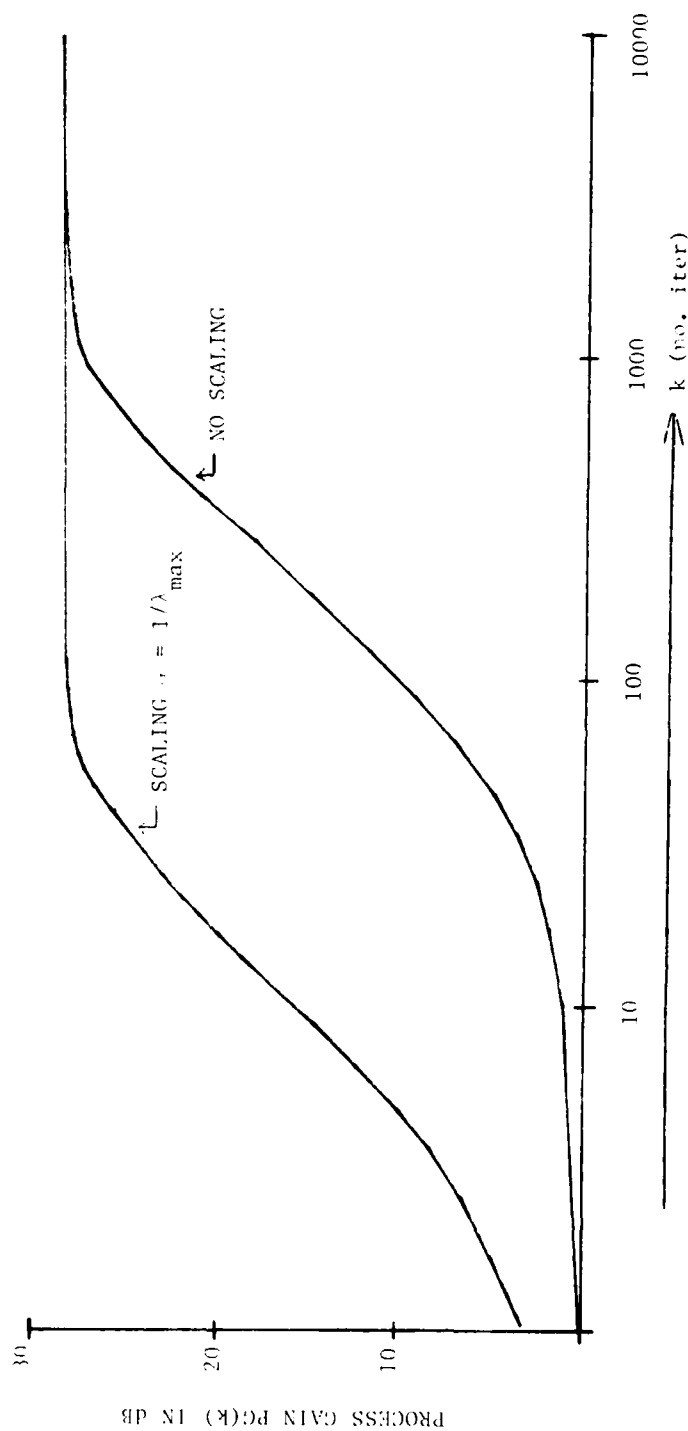
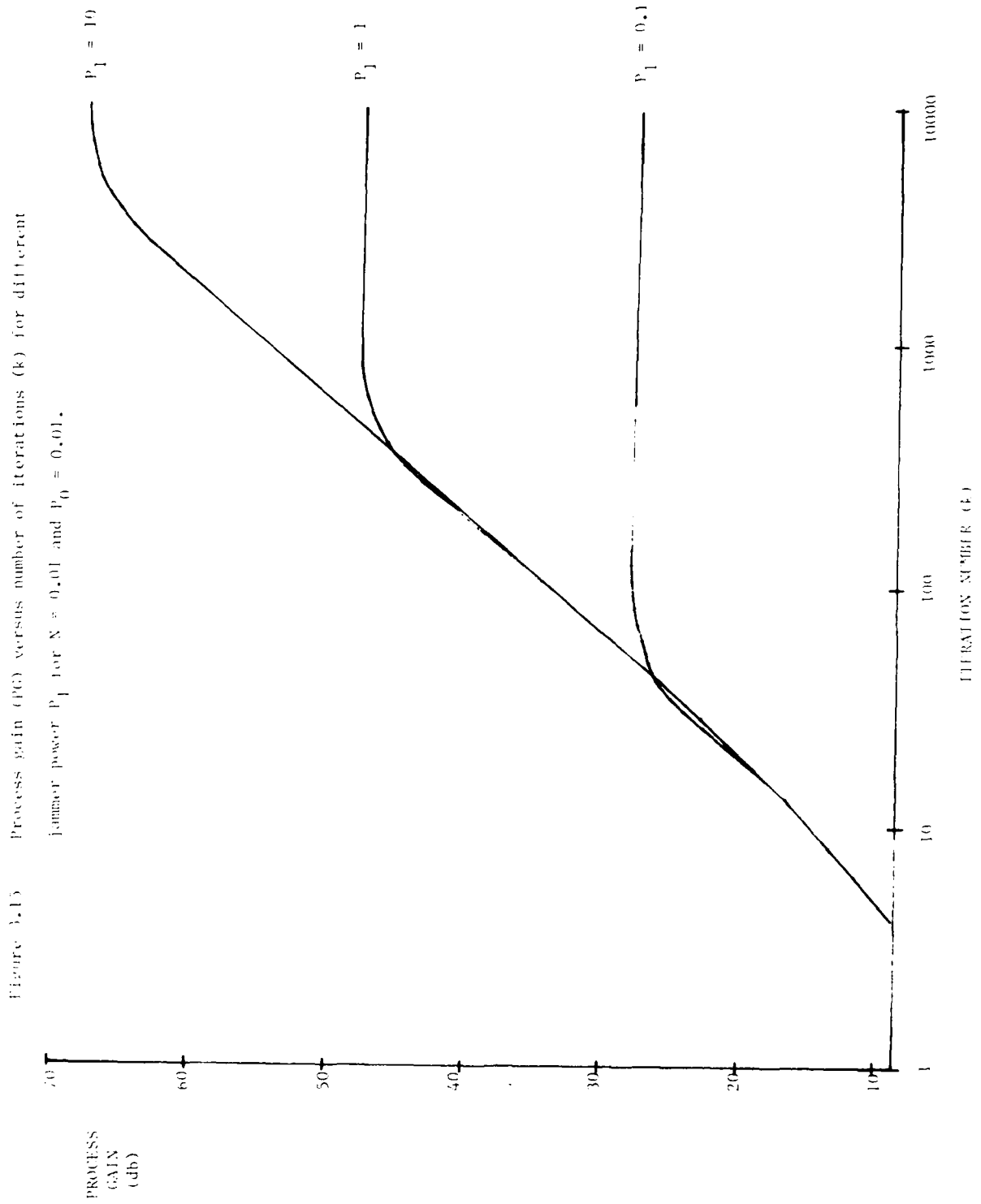


Figure 3.14 Effect of scaling on process gain PG as a function of the number of iterations.

$$P_1 = 0.01, P_0 = 0.001, N = 0.001$$



PROCESS GAIN (db)

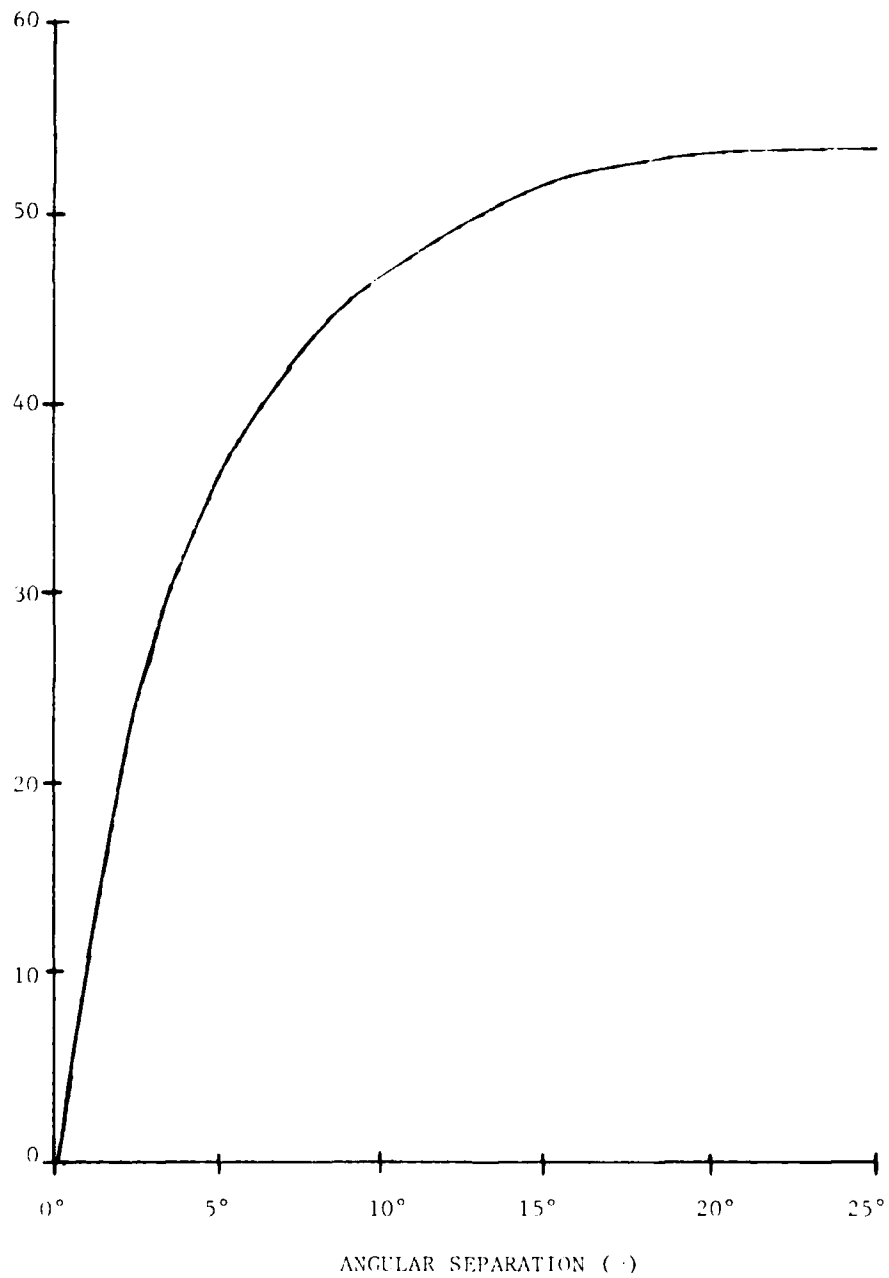


Figure 3.16 Process gain (PG) obtainable (after 1000 iterations) for different angular separations between the target and jammer for $P_1 = 1$, $P_0 = 0.01$ and $N = 0.01$.

and the jammer. These data show that the null depth or PG obtainable decreases as θ decreases as expected and that the system provides adequate nulls and SNRs for $\theta > 10^\circ$. Finally, we investigated the performance of the IOP system with multiple jammers ($M = 1-7$) at 5° increments centered at 30° with $P_1 = 1.0 \gg P_0 = P_r = 0.01$. We found PG in steady state to decrease as M increases as expected. The initial $E_k(\theta)$ patterns were calculated and plotted for the seven jammer case. The $E(\theta)$ plots showed clear nulls at the noise sources as desired.

These simulations verify the use of the IOP and the simulator and the importance of proper selection of θ and its effect on the performance of the APAR processor through SNR, PG and number of iterations k necessary.

3.6 EXPERIMENTS

We now consider one detailed IOP lab demonstration experiment. In Sect. 3.6.1 we describe the noise source scenario and the associated 2D-1D antenna model used. The case of an APAR processor with adaptivity in space and time operating on complex-valued data is chosen. In Sect. 3.6.2, expressions for \underline{s} , \underline{M} , and \underline{w} and the associated optical masks are described. In Sect. 3.6.3, experimental IOP laboratory data on the optically computed weights is presented and compared to the exact values obtained by simulation. The performance of the IOP system for APAR is best obtained by computing the resultant $E(\theta)$ adaptive antenna pattern produced by applying the optically computed weights to the antenna. The SNR obtained and the depth of the location of the nulls produced by the adaptive system is the performance criteria used. To assess the accuracy of the optical IOP system, the theoretical SNR and null depths for a perfect processor and for a processor with the IOP errors were digitally calculated. The additive and multiplicative noise error source values measured for the IOP (Sect. 3.5) were included in the advanced simulator model. The $E(\theta)$, SNR and null depths as well as PG expected from the IOP with its given accuracy are computed and compared to the experimentally obtained data.

3.6.1 Noise Scenario

The APAR noise scenario chosen was one noise source (jammer) with power $P_1 = 1.0$ located at $\theta_1 = 0^\circ$ with a velocity $v_1 = 0$ relative to the antenna. The receiver noise power was $P_r = 1.0$ and the signal power $P_0 = 0.1$ and thus $P_0 \ll P_1$. We assume that the signal is at $\theta_0 = 45^\circ$ with $v_1 = 250$ relative to the antenna. A 5×5 antenna model with $N = 5$ taps (five antenna elements) and $N' = 5$ time taps for the signal received at each antenna element was used. We assumed a two element antenna with spacing $d = \lambda/2 = 0.5$ or $\lambda = 1$ and time delays between taps of $\Delta t = T = 0.0005 = \lambda/4v_{\max}$ corresponding to a maximum target velocity of 500 as described in Sect. 3.4. The four adaptive elements chosen from this 2-D antenna model were two adjacent antenna elements with two adjacent time taps from each. The associated 2-D and 1-D mapping is shown in Fig. 3.17.

3.6.2 M And Mask Determination

The covariance matrix

$$\underline{M} = \underline{z}(t) \underline{z}^T(t) \quad (3.50)$$

with elements

$$m_{ij} = \int z_i(t) z_j^*(t) dt \quad (3.51)$$

was computed in our simulator assuming a narrow-band jammer, uncorrelated receiver noise and signal power P_0 much less than the noise jammer power P_1 . The matrix \underline{M} obtained (complex-valued elements) is

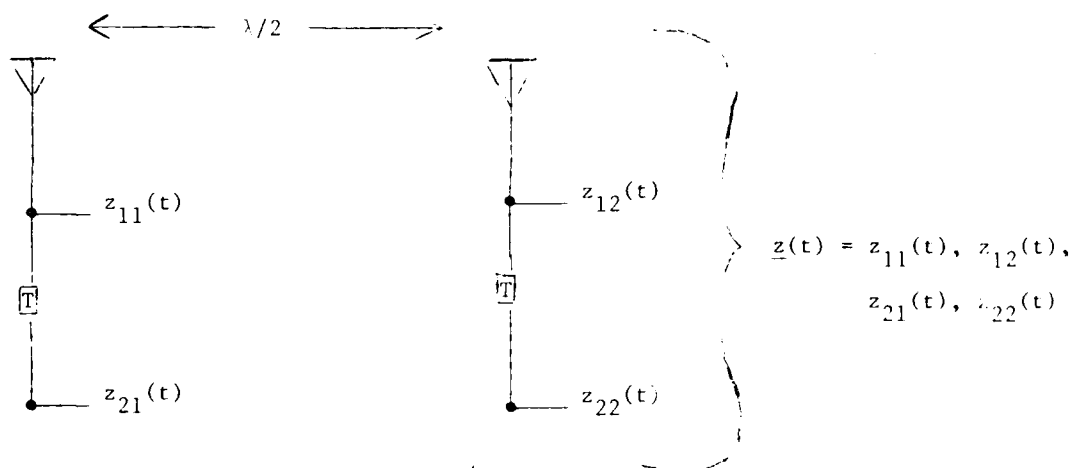


Figure 3.17 2-D to 1-D antenna model used for scenario.

$$\underline{M} = \begin{bmatrix} 2.0 & 1.0 & 1.0 & 1.0 \\ 1.0 & 2.0 & 1.0 & 1.0 \\ 1.0 & 1.0 & 2.0 & 1.0 \\ 1.0 & 1.0 & 1.0 & 2.0 \end{bmatrix} \quad (3.52)$$

The largest eigen-value of \underline{M} was then determined as

$$\lambda_{\max} = \sqrt{\sum_{i=1}^4 \sum_{j=1}^4 m_{ij} m_{ij}^*} = 7.48 = \lambda_1. \quad (3.53)$$

The acceleration factor used in the IOP was chosen as

$$\alpha = 1/\lambda_{\max} \approx 0.13. \quad (3.54)$$

The complex-valued \underline{M} in (3.52) was then partitioned into its real and imaginary bipolar part, \underline{M}_r and \underline{M}_i and these were arranged as

$$\underline{M} = \begin{bmatrix} \underline{M}_r & -\underline{M}_i \\ \underline{M}_i & \underline{M}_r \end{bmatrix} = \begin{bmatrix} 2 & 1 & 1 & 1 & | & 0 & 0 & 0 & 0 \\ 1 & 2 & 1 & 1 & | & 0 & 0 & 0 & 0 \\ 1 & 1 & 2 & 1 & | & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 2 & | & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & | & 2 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & | & 1 & 2 & 1 & 1 \\ 0 & 0 & 0 & 0 & | & 1 & 1 & 2 & 1 \\ 0 & 0 & 0 & 0 & | & 1 & 1 & 1 & 2 \end{bmatrix} \quad (3.55)$$

The optical mask \underline{B} was obtained from $\underline{H} = \underline{I} - \underline{M}$ by scaling \underline{H} by dividing its elements by $(\bar{h} - \underline{h}) = 2$ and then biasing this up by addition of $\underline{h}/(\bar{h} - \underline{h}) = 0$ to each element, where \bar{h} and \underline{h} are the maximum and minimum values of the elements of \underline{H} . The optical mask is thus described by

$$\underline{B} = \begin{bmatrix} 1 & 1/2 & 1/2 & 1/2 & 0 & 0 & 0 & 0 \\ 1/2 & 1 & 1/2 & 1/2 & 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 1 & 1/2 & 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 1/2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1/2 & 1/2 & 1/2 \\ 0 & 0 & 0 & 0 & 1/2 & 1 & 1/2 & 1/2 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 & 1 & 1/2 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 & 1/2 & 1 \end{bmatrix} \quad (3.56)$$

The optical mask corresponding to (3.56) was formed by a computer plotter and accurately photoreduced. A photograph of it is shown in Fig. 3.18.

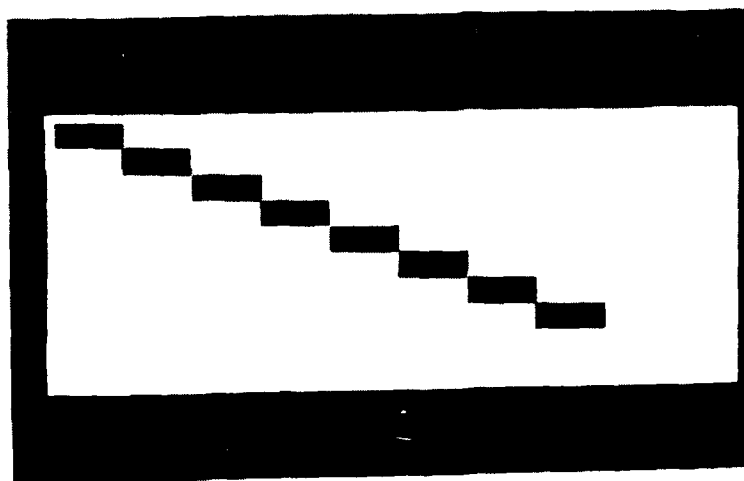


Figure 3.18 Photograph of the optical mask.

The steering vector for $\alpha_0 = 0^\circ$ and $v_0 = 0$ is

$$\underline{s}^* = \begin{bmatrix} -0.84 + j0.62 \\ 0.99 + j0.27 \\ 0.62 + j0.84 \\ 0.27 - j0.99 \end{bmatrix} \quad (3.57)$$

It is easily decomposed into its real and imaginary parts as

$$\underline{v} = \begin{bmatrix} \underline{s}_r^* & \underline{s}_i^* \end{bmatrix}^T = \begin{bmatrix} -0.84 & 0.99 & 0.62 & 0.27 & 0.62 & 0.27 & 0.84 & -0.99 \end{bmatrix}^T. \quad (3.58)$$

This vector was electronically entered into the micro-processor electronic feedback system of the IOP with the necessary scale, bias and acceleration factors.

3.6.3 Experimental Data

The above vector and mask elements were entered into the IOP system. The initial starting value used for \underline{w} (the initial LED outputs at the first iterative cycle) was

$$\underline{w}_{init, k=0} = [0]^T. \quad (3.59)$$

The IOP computes

$$\underline{x}(k) = \underline{x}(k-1) + \epsilon [\underline{y} - \underline{H} \underline{x}(k-1)] \quad (3.60)$$

for each iteration k where all parameters are bipolar. With eight elements for \underline{y} and \underline{x} and an 8×8 mask for \underline{H} , the eight \underline{x} outputs at iteration k are

$$\underline{x}(k) = [\underline{x}_1(k), \dots, \underline{x}_8(k)]^T, \quad (3.61)$$

where \underline{x}_1 - \underline{x}_4 and \underline{x}_5 - \underline{x}_8 correspond to \underline{s}_r and \underline{s}_i respectively because of the data format used. The complex-valued output weights $\underline{w}(k)$ at iteration k ,

$$\underline{w}(k) = [\underline{w}_{11}(k), \underline{w}_{12}(k), \underline{w}_{21}(k), \underline{w}_{22}(k)]^T, \quad (3.62)$$

to be applied to the four time and space taps in Fig. 3.17 are computed from the x_1 - x_8 outputs in (3.61) as

$$w_{11}(k) = x_1(k) + jx_5(k) \quad (3.63a)$$

$$w_{12}(k) = x_2(k) + jx_6(k) \quad (3.63b)$$

$$w_{21}(k) = x_3(k) + jx_7(k) \quad (3.63c)$$

$$w_{22}(k) = x_4(k) + jx_8(k). \quad (3.63d)$$

3.6.4 Analysis

In an actual system, the weights would be applied to the $\underline{z}(t) = [z_{11}(t), z_{12}(t), z_{21}(t), z_{22}(t)]^T$ antenna outputs and the results added to form the antenna output

$$v_{out}(t,k) = \sum_{i=1}^2 \sum_{j=1}^2 z_{ij}(t)w_{ij}(t). \quad (3.64)$$

In Fig. 3.19, we show the experimentally obtained outputs corresponding to the $\underline{x}(k)$ data obtained from the IOP after iterations 0, 5, 50. These scope outputs are the $\underline{x}(t) = [x_1(k), \dots, x_8(k)]^T$ outputs from (3.60) at $k = 0, 5$, and 50. The experimental values and the exact values predicted by theory are listed in Table 3.8. From the steady state ($k = 50$) values given, we find an rms error

$$\epsilon = \left\{ [\underline{x} - \underline{x}(k)]^T [\underline{x} - \underline{x}(k)] \right\}^{1/2} \quad (3.65)$$

of 0.26 rms or approximately 0.1 of a division on the scope displays in Fig. 3.19.

TABLE 3.8
ANTENNA WEIGHTS IN STEADY STATE (k = 50)

Parameter	Experimental	Theoretical (Exact)
x_1	-0.9	-1.0
x_2	0.65	0.76
x_3	0.35	0.4
x_4	0.1	0.05
x_5	0.3	0.46
x_6	0.2	0.13
x_7	0.75	0.65
x_8	-1.1	-1.1

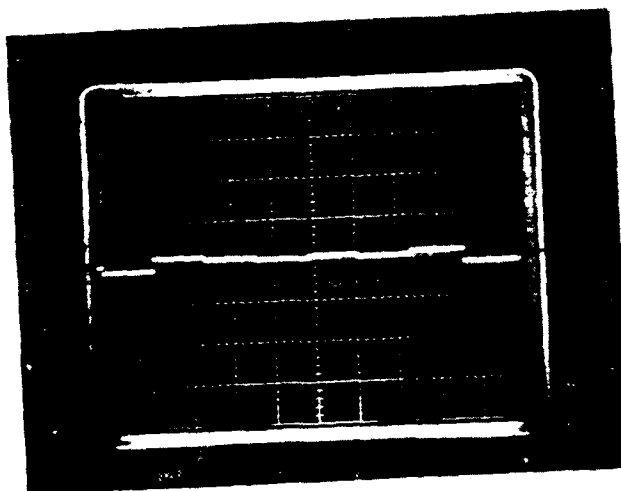
We then computed the adaptive antenna pattern $E(\hat{u}, v)$ produced by the theoretical (exact) data and the experimental data, as our comparison measure we use the antenna SNR

$$\text{SNR} = \frac{P_0 |E(\hat{u}_0, v_0)|^2}{P_1 |E(\hat{u}_1, v_1)|^2} \quad (3.66)$$

The experimental data gave an SNR = 14.79 dB, whereas the exact theoretical values gave an SNR = 14.95 dB. These are in excellent agreement. We then used the advanced simulator with the additive and multiplicative system errors included and found an SNR = 15 ± 0.75 dB. Thus, the experimental data is well within the range predicted by the measured system errors and is in excellent agreement with the theoretical values, with an error between experiment and exact theory of only 0.16 dB.

$\underline{x}(0)$

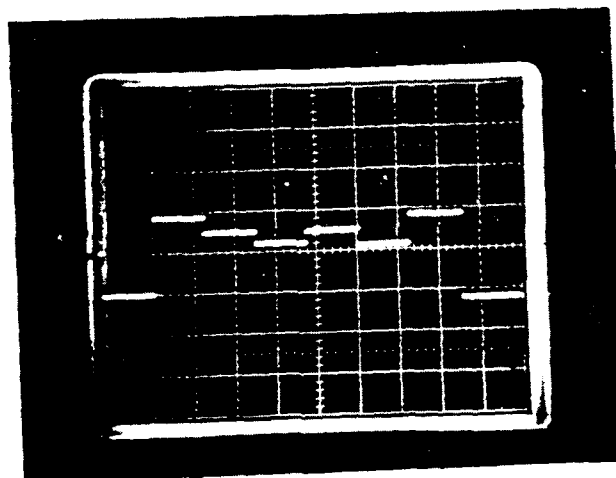
+1 —
 0 —
 -1 —



$\underline{x}(0)$
 INITIAL
 ESTIMATE

$\underline{x}(5)$

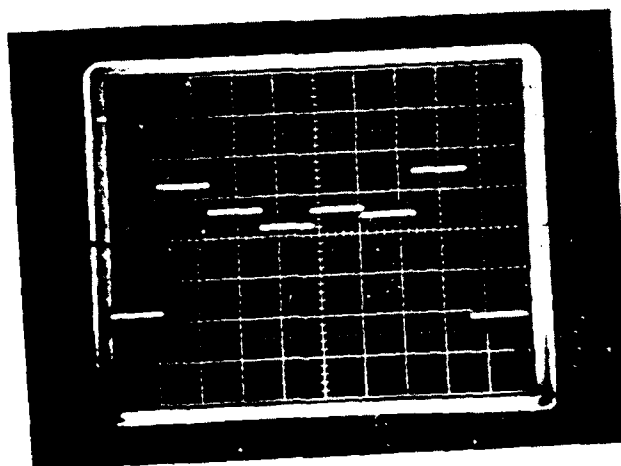
+1 —
 0 —
 -1 —



$\underline{x}(5)$
 AFTER 5
 ITERATIONS

$\underline{x}(50)$

+1 —
 0 —
 -1 —



$\underline{x}(50)$
 STEADY STATE
 (50 ITERATIONS)

Figure 3.19 IOP outputs after iterations 0, 5, 50 (steady state). Each division vertically is 0.2.

3.7 SUMMARY AND CONCLUSION

The IOP concept for APAR processing has proven to be most excellent. The highlights of our recent work on this approach to APAR processing are summarized below.

- (1) Design and fabrication of a new IOP system with fiber optic interconnections, pulse width modulation and a microprocessor electronic feedback system (Sect. 3.3).
- (2) Use of a new technique to allow the IOP to operate on complex-valued data (Sect. 3.2.4).
- (3) Development and demonstration of a new IOP simulator, antenna model, and 2-D to 1-D mapping with space and time adaptivity and including IOP error sources (Sect. 3.4).
- (4) Demonstration and measurement of the excellent accuracy of the IOP system to be less than 0.8% and with less than 1% error in the antenna SNR obtained (Sect. 3.5.1).
- (5) Development and demonstration of a new technique to ensure convergence to the steady state solution and fast convergence of the IOP system (Sect. 3.5.2).
- (6) Real-time demonstration and analysis of the laboratory IOP system for antenna adaptivity in space and time (Sect. 3.6).

The weights and antenna SNRs obtained experimentally were within 1% of the theoretical limit. This excellent system performance in the IOP laboratory system fabricated represents a major new optical data processing architecture that appears to be quite attractive and useful for APAR processing and other applications.

CHAPTER 4 WAVELENGTH DIVERSITY PROCESSOR

4.1 INTRODUCTION

In this chapter, we consider the use of color to provide an added dimension to optical processors with attention to their applicability to various aspects of the APAR problem. In Sect. 4.2, we provide a general overview of the wavelength diversity processor. Specific details on the design of the system we assembled are provided in Sect. 4.3. The original reason [6] for considering the use of a wavelength diversity processor was to allow complex-valued data to be efficiently processed in our non-coherent processor [5]. In Sect. 4.4, we review and summarize the various possible techniques for complex-valued data processing in non-coherent systems. The specific use of the wavelength diversity processor in processing complex-valued data is then addressed in Sect. 4.5. A new technique is introduced in Sect. 4.5 that produces the complex output directly, without the post processing required in the system we previously described [5].

In subsequent sub-sections, we consider other advanced aspects of this wavelength diversity processor for various APAR operations. Included are the system's use as a multi-channel 1-D vector-matrix convolver (Sect. 4.6), a matrix-matrix multiplier (Sect. 4.7), a matrix inverter (Sect. 4.8) and a covariance matrix computer (Sect. 4.9). Experimental verification and demonstration are included for the applications in Sects. 4.5 and 4.7.

4.2 GENERAL PROCESSOR DESCRIPTION

The general schematic for the wavelength diversity APAR processor is shown in Fig. 4.1. The input at P_A can contain L linear laser diode (LD) or light emitting diode (LED) arrays, [or a color-separated wideband light source and 2-D spatial light modulator (SLM) system],

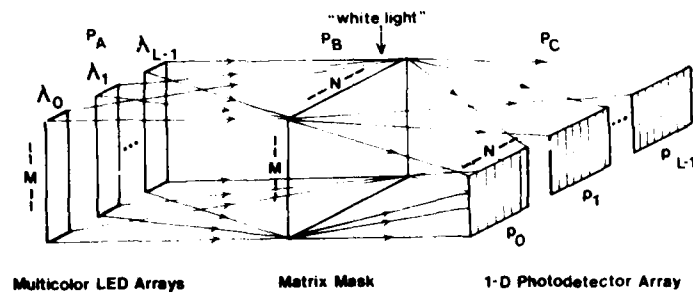


Fig. 4.1 Schematic of the wavelength diversity APAR processor.

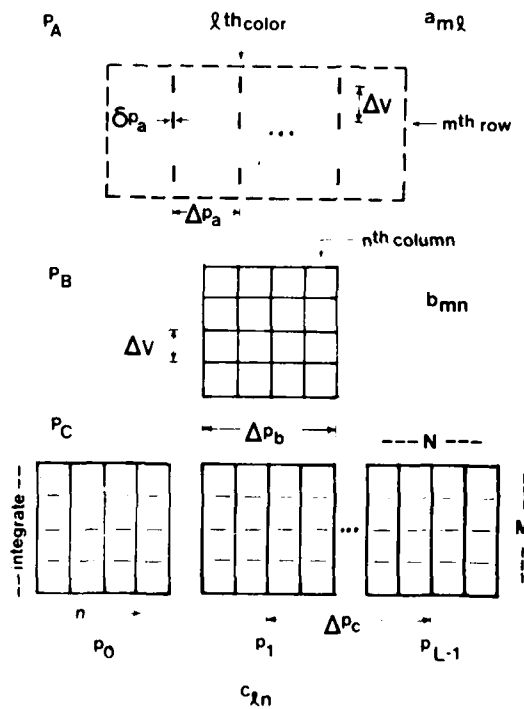


Fig. 4.2 Format and notation for P_A and P_B patterns and for the data incident on P_C .

each emitting at different wavelengths $\lambda_i = \lambda_0$ to λ_{L-1} . Advanced LD technology [23, 24] and advances in optical communications [25] appear to allow fabrication of such an input LD array. In the system shown, the output light from each LD array is spread horizontally (to uniformly illuminate each row m of the mask at P_B) and imaged vertically.

We denote the output from P_A by a_m , and the transmittance of the mask at P_B by b_{mn} . Leaving P_B , we find the product $b_{mn} a_m$. Note that the light from each linear LD array is combined in P_B , thus superimposing all L input colors λ_i and hence providing essentially white light illumination of P_B horizontally. Behind P_B , we place a grating that separates the different input wavelengths λ_i horizontally in P_C with all λ_0 light going to output P_0 , all λ_1 light to P_1 , etc. at P_C . Plane P_B is imaged onto P_C . The linear detector arrays at P_C have sufficient height so that the rows of P_B are imaged within each detector element at P_C . This provides the summation over m of the P_B outputs. The final P_C output is thus

$$c_{in} = \sum_m^M b_{mn} a_m \quad (4.1)$$

where P_0 in Fig. 2.1 is c_{0n} , P_1 is c_{1n} , etc. We will use a , b and c to denote optical values and x , H , and y to denote data values in this chapter.

The P_C output plane thus contains L linear 1-D detector arrays each with N elements, or equivalently one $L \cdot N$ element array. This 1-D output detector topology is preferable to use of 2-D arrays, both from fabrication considerations and from the ease with which appropriate horizontal spacings between the P_0 , P_1 etc. sections of the P_C array can be achieved (compared to the ease in obtaining the necessary vertical separations between rows of a 2-D array or between vertically stacked 1-D arrays).

The general format for P_A , P_B and the data incident on P_C is shown in Fig. 4.2. As shown in Fig. 4.2, P_A has L elements horizontally and M elements vertically. We denote the width of each element (LD width) by Δp_a , their center-to-center horizontal spacing by Δp_a and their vertical spacings by Δv . The horizontal size of the P_B mask is denoted by Δp_b and the vertical (as well as horizontal) size of an element of P_B is chosen as Δv (this allows 1:1 imaging vertically of P_A onto P_B). The pattern incident on P_C is shown in the bottom of Fig. 4.2. It contains L 2-D patterns each of $M \times N$ elements. Each of these $M \times N$ patterns incident on P_C is the product of one column of A at one λ and the P_B mask. The center-to-center separation between each $M \times N$ array is Δp_c . The P_C detector with its long height integrates or sums the M outputs in each column at P_C . In Sect. 4.3, we will relate the Δp and δp parameters in Fig. 4.2 to the system's design factors.

In Fig. 4.3, we show the functional block diagram of the wavelength diversity processor of Fig. 4.1. The system thus forms at its output the products of L input vectors \vec{a}_λ and the matrix B .

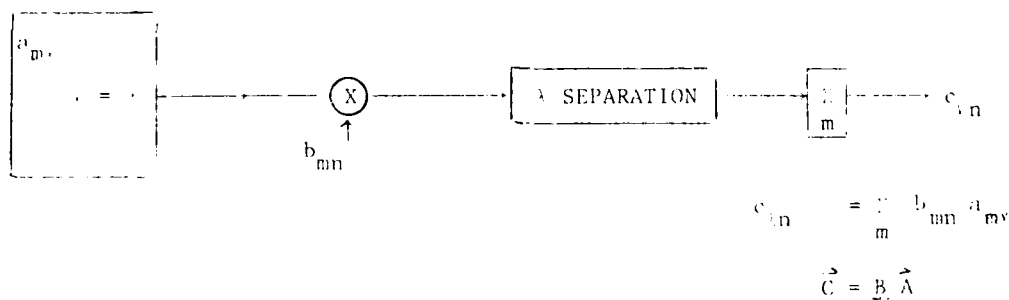


Fig. 4.3 Functional block diagram of the wavelength diversity APAR processor.

Before describing the assembled system and its parameterization and design, we discuss an advanced version of the system in which three matrix masks \underline{B}_0 , \underline{B}_1 and \underline{B}_2 (with center-to-center spacing Δp_c) are placed at P_B (Fig. 4.4). Consider the case of three linear LD input arrays \vec{a}_0 , \vec{a}_1 and \vec{a}_2 each emitting at a different λ (Fig. 4.4 top). Each LD's output is spread horizontally to illuminate all of P_B and each linear LD array is imaged vertically onto P_B . The P_B output is again passed through a grating that separates the colors. The spacings between the P_B masks is Δp_c , the same as the separation between each P_i output at P_C . With proper choice of spacings and grating frequencies, the pattern incident on P_C is as shown in Fig. 4.4 (bottom).

The P_C output pattern (Fig. 4.4, bottom) obtained for this case is best seen by considering each \vec{a}_i input separately. The \vec{a}_0 input at λ_0 multiplies all three \underline{B} masks at P_B and produces $\underline{B} \vec{a}_0$ at P_C . Because the three masks at P_B are spaced by Δp_c and because P_B is imaged horizontally onto P_C , the $\underline{B} \vec{a}_0$ output at P_C consists of three terms: $\underline{B}_0 \vec{a}_0$, $\underline{B}_1 \vec{a}_0$ and $\underline{B}_2 \vec{a}_0$ separated horizontally at P_C by Δp_c . We now consider the \vec{a}_1 input at P_A at λ_1 . The P_C output resulting from this input is $\underline{B} \vec{a}_1$ and again contains three terms: $\underline{B}_0 \vec{a}_1$, $\underline{B}_1 \vec{a}_1$ and $\underline{B}_2 \vec{a}_1$ each separated by Δp_c . By proper choice of $\Delta p = \lambda_1 - \lambda_0$ and the frequency of the grating, the λ difference causes the $\underline{B} \vec{a}_1$ pattern at P_C to be displaced from the $\underline{B} \vec{a}_0$ pattern at P_C by Δp_c which is also the width of an individual $\underline{B}_i \vec{a}_i$ pattern (with 1:1 imaging assumed from P_B to P_C). To achieve a Δp_c shift at P_C due to λ , we must satisfy

$$\Delta p_b = \Delta p_c = \Delta \lambda f_{010}^d$$

where the terms involved are discussed in Sect. 4.3. The P_C patterns are displaced due to λ .

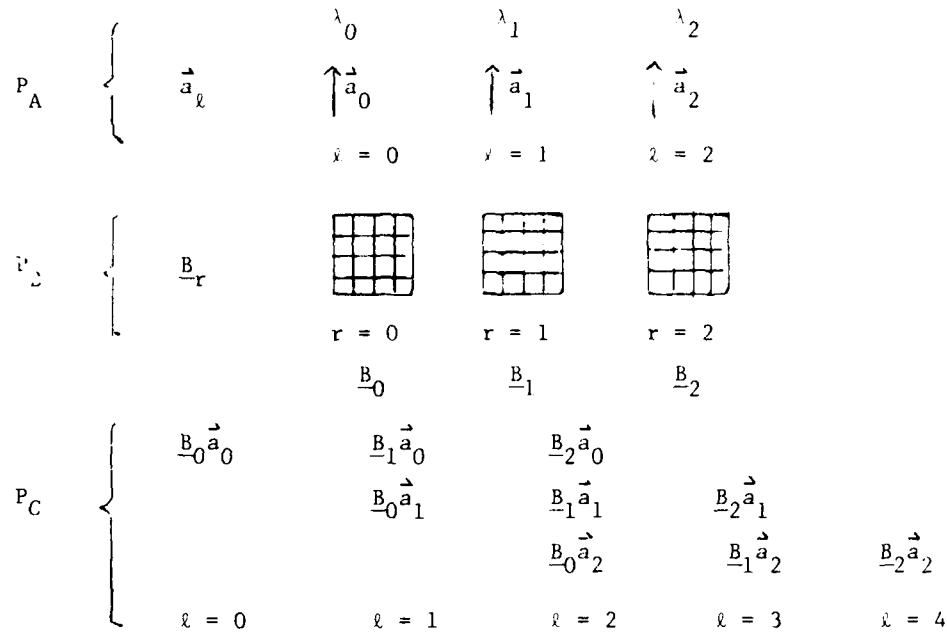


Fig. 4.4 General patterns for the wavelength diversity APAR processor.

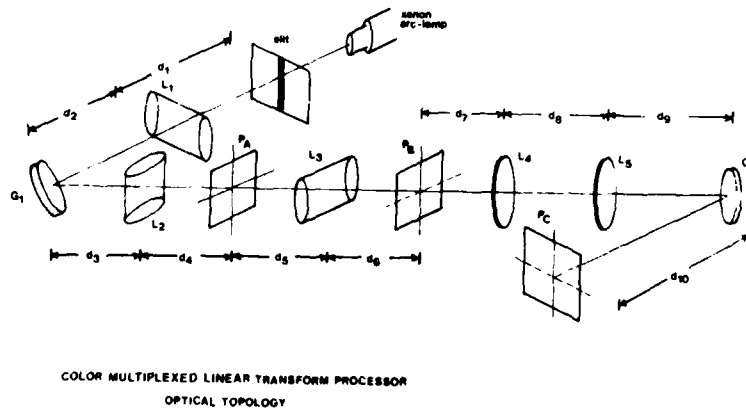


Fig. 4.5 Schematic of the experimental wavelength diversity APAR processor.

Continuing with our discussion of the anatomy of the P_C pattern in Fig. 4.4 (bottom), the P_C output $B \vec{a}_2$ due to the \vec{a}_2 input at λ_2 is displaced by Δp_C from the $B \vec{a}_1$ output. In Fig. 4.4 (bottom), we show the separate $B \vec{a}_i$ outputs at P_C separated vertically to clarify what the contents of P_C are. In actuality, the three P_C outputs shown are added vertically on the output detector to obtain the desired output.

4.3 EXPERIMENTAL WAVELENGTH MULTIPLEXED PROCESSOR

The wavelength diversity APAR processor that we assembled is shown schematically in Fig. 4.5. Since linear LD source arrays were not available to us, we produced the equivalent P_A pattern using an arc lamp source and mask. Referring to Fig. 4.5, white light from a 1000 watt Xenon arc lamp was separated into different colors by reflection from a blazed grating G_1 at $f_0 = 1200$ line/mm spatial frequency. Lens L_1 images the slit vertically onto P_A .

$$\frac{1}{d_1} + \frac{1}{d_2} + \frac{1}{d_2} + \frac{1}{d_3} = \frac{1}{f_{cl}} \quad (4.2)$$

whereas L_2 images the slit horizontally across P_A

$$\frac{1}{d_1 + d_2 + d_3} + \frac{1}{d_4} = \frac{1}{f_{c2}}. \quad (4.3)$$

Thus incident on P_A , we find separate wavelengths or colors of light $\lambda_x = \lambda_0, \lambda_1$ etc. at different horizontal locations across P_A . At P_A we place a mask whose transmittance in $x = \lambda$ and $y = m$ corresponds to the desired a_m pattern.

Lens L_3 performs 1:1 imaging of P_A vertically onto P_B ,

$$\frac{1}{d_5} + \frac{1}{d_6} = \frac{1}{f_{c3}}. \quad (4.4)$$

Lens L_2 also images G_1 horizontally onto P_B

$$\frac{1}{d_3} + \frac{1}{d_4 + d_5 + d_6} = \frac{1}{f_{c2}}. \quad (4.5)$$

This effectively recombines the different λ_x colors of light incident on P_A from G_1 at P_B so that P_B is illuminated with essentially white light (i.e. all λ_x from all L sources is spread horizontally across P_B as in Fig. 4.1).

If a grating G_1 of spatial frequency f_0 is illuminated with light of wavelength λ , then the first-order diffracted spot in a plane P_A a distance d from G_1 lies a distance p from the on-axis location, where

$$p = f_0 \lambda d. \quad (4.6a)$$

In the system in Fig. 4.5, the spatial frequency at G_1 is f_0 and the spatial frequency at L_2 is f'_0 , where

$$\frac{f_0}{d_1 + d_2} = \frac{f'_0}{d_1 + d_2 + d_3} \quad (4.7)$$

since $f_0 = f'_0$ if $d_3 = 0$ and the effective frequency f'_0 should be used in (4.6a) with $d = d_4$. Solving (4.7) for f'_0 at the plane of L_2 and substituting this expression for f_0 in (4.6a) with $d = d_4$, we can relate input λ , horizontal displacement p in P_A and lens spacings by

$$p = -f_0 d_4 (d_1 + d_2) / (d_1 + d_2 + d_3). \quad (4.6b)$$

Rewriting (4.6b) in terms of the wavelength separation $\Delta\lambda = \lambda_{\ell+1} - \lambda_{\ell}$ between adjacent vectors and the spatial separation δp_a (see Fig. 4.2) between the input vectors in P_A , we obtain

$$\delta p_a = (-\lambda) f_0 d_4 (d_1 + d_2) / (d_1 + d_2 + d_3). \quad (4.6c)$$

The P_A masks used were 5×5 element binary arrays. They were drawn on Amberlith A3A masking film and then photoreduced 20X onto Kodak 1A high resolution plates. For the final P_A inputs, $\delta p_a = 2.5$ mm and $\delta p_a = 0.15$ mm and $\Delta\lambda = 1$ nm. All cylindrical lenses used had focal lengths $f_c = 100$ mm. Satisfying all focusing and imaging conditions, all d_i values in (4.2)-(4.5) are found and $\Delta\lambda = 48$ nm in (4.6) results. This value appears to be typical of the $\Delta\lambda$ difference to be expected between LD arrays of different colored light [26]. With $\delta p_a = 0.15$ mm, the spectral width of each λ_i is $\Delta\lambda = 0.64$ nm. The value lies between the typical spectral line width of an LED (30-40 nm) and a LD (0.01 nm) [24].

We now return to the details of the rest of the system in Fig. 4.5. Lenses L_4 and L_5 image P_B with 1:1 magnification onto P_C ,

$$d_8 = f_{s4} + f_{s5} \quad (4.8a)$$

$$d_7 = d_9 + d_{10} = f_{s4} = f_{s5}, \quad (4.8b)$$

where $f_s = 381$ mm is the focal length of all spherical lenses used. This 1:1 imaging from P_B to P_C maintains matched P_B and P_C plane sizes, with the separations between $\underline{B}_n^{\underline{a}}$ outputs equal to the separation Δp_c between the \underline{B}_n mask patterns. It is thus necessary to ensure that the separations at P_C between the different λ_ℓ wavelengths (of separation $\Delta\lambda$) also equals Δp_c . The grating G_2 performs the separation of the λ_ℓ colors at P_C . These colors were initially spatially separated horizontally at P_A by Δp_a as in (4.6c) and later recombined at P_B by condition (4.5). The separation Δp_c at P_C between the different λ_ℓ is related to their wavelength separation $\lambda_{\ell+1} - \lambda_\ell$ by

$$\Delta p_c = \Delta\lambda f_0 d_{10} . \quad (4.9)$$

Since each vector in \underline{a} at P_A is not just one λ but contains a range $\delta\lambda$ of wavelengths (due to the finite slit width δp_a used (Fig. 4.2) as well as due to practical $\delta\lambda$ ranges for LDs), the δp_a spatial spread due to a $\delta\lambda$ is given by (4.6c) with $p = \delta p_a$ and $\lambda = \delta\lambda$, or $\delta p_a = f_0' (\delta\lambda)d$. The P_C output will also have a spatial spread or resolution δp_c due to δp_a given by (4.6a) with $d = d_{10}$, and $p = \delta p_c$. The horizontal image resolution in P_C due to δp_a is thus

$$\delta p_c = \delta p_a (d_{10}/d_4) (d_1 + d_2 + d_3)/(d_1 + d_2) \quad (4.10)$$

Experimentally $\delta p_c \leq 0.1$ mm was obtained. With LD sources, typical spectral line widths are 0.01 nm; therefore we expect about a 10X improvement in horizontal image resolution.

When the actual system of Fig. 4.5 is used with multiple linear LD input arrays at P_A , lens L_2 is placed f_{c2} to the right of P_A . This collects all light from the LDs and images each LD column at P_A horizontally across P_B , thereby illuminating P_B with white light.

4.4 REVIEW OF COMPLEX AND BIPOLAR DATA HANDLING

In this section, we review the major techniques by which complex-valued data and bipolar data can be handled, represented, processed and operated upon in a non-coherent vector-matrix processor of the type used.

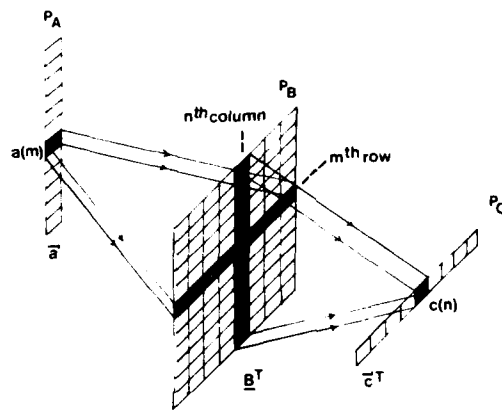


Fig. 4.6 Schematic of basic vector-matrix non-coherent APAR processor.

Consider first the system of Fig. 4.6 whose input at P_A is a 1-D vector \underline{a}_m . With a mask $b_{m,n}$ at P_B , the P_C output is

$$c_n = \sum_{m=1}^M b_{m,n} a_m, \quad (4.11)$$

where M elements at P_A , $N \times M$ at P_B and N at P_C are assumed. The P_C output in (4.11) is the vector matrix product

$$\underline{\hat{c}} = \underline{B} \underline{\hat{a}} \quad (4.12)$$

where $(\underline{})$ denotes a vector and $(\underline{})$ a matrix. Since such non-coherent processors

can operate only on positive valued data, new techniques are necessary when the data values are bipolar or complex (the latter is the case in APAR). We first consider two methods to process bipolar data and then two techniques for complex-valued data. In Sect. 4.5, we describe a new and preferable method for complex-valued data processing using the wavelength diversity processor. In Chapter 3, we consider a new method when multiple wavelength sources are not available. This latter technique is the one used in our IOP system, the one discussed in Sect. 4.5 is the one used in the wavelength-diversity processor (the subject of the present chapter).

In the first bipolar-valued data processing method considered, we bias x_m and $b_{m,n}$ data as $(1/2 + x_m)$ where $-1/2 \leq x_m \leq 1/2$, so that the recorded data a_m and $b_{m,n}$ satisfy $0 \leq a_m, b_{m,n} \leq 1$. The output from detector n at P_C will then be the sum of the light leaving the n -th column of P_B or

$$c_n = \sum_m (1/2 + b_{m,n}) (1/2 + x_m). \quad (4.13)$$

The correct y_n output (4.11) can be obtained from c_n by

$$y_n = c_n - \sum_m (a_m/2 + b_{m,n}/2) + M/4. \quad (4.14)$$

The $\sum_m a_m/2$ and $M/4$ terms in (4.14) can be optically computed by addition of one extra column to B with uniform transmittance of one-half. At the correct detector output in P_C , we obtain $\sum_m a_m/2 + M/4$, which can thus be subtracted as a constant from all observed c_n outputs in an electronic post processor. Computing $\sum_m b_{m,n}/2$ could also be performed optically in one iteration with all $a_m = 1/2$. The c_n output in P_C is then $\sum_m b_{m,n}/2 + M/4$. This technique reduces the useable dynamic range of the LED input, the B mask and the output detector, but it requires electronic post processing plus an extra iteration of the optical system (or electronic computation of $\sum_m b_{m,n}$ for each new B). This technique only

increases the space bandwidth (SBW) of B to $(N + 1)(M + 1)$ from NM and is thus attractive.

In the second method of bipolar data processing we consider, each component is divided into its positive $()^+$ and negative $()^-$ parts. The output from P_A is thus $\begin{bmatrix} \vec{x}_m^+ \\ \vec{x}_m^- \end{bmatrix}^T$ where the top half of the LED source array is fed with \vec{x}^+ and the lower half with \vec{x}_m^- . Note that this scheme now requires a $2M$ element input LED array. The B mask at P_B is similarly decomposed into four sections (increasing its SBW by a factor of 4 from MN to $2M \cdot 2N$) as in (4.15). The c_n output at P_C is thus

$$\begin{bmatrix} \vec{y}_n^+ \\ \vec{y}_n^- \end{bmatrix} = \begin{bmatrix} H^+ & H^- \\ H^- & H^+ \end{bmatrix} \begin{bmatrix} \vec{x}_m^+ \\ \vec{x}_m^- \end{bmatrix} \quad (4.15)$$

A $2N$ element linear output detector is thus needed. The \vec{y}_n^+ outputs appear on the first N detector elements and the \vec{y}_n^- outputs on the last N detector elements. The increased SBW of P_A-P_C is a disadvantage of this system. However, it requires no electronic post processing and does not reduce system dynamic range.

Specific quantification of the dynamic range of the a_m , $b_{m,n}$ and c_n elemental values and the P_A-P_C component specifications (SBW and dynamic range) for exact scenarios is needed to enable a decision between these two bipolar data processing schemes to be made. We now consider two techniques for complex-valued data processing.

An attractive method for performing vector-matrix multiplications on complex-valued data was described by Bond [27] for CCD processing and Goodman, et al. [28] for optical processing. It was then applied to a vector-matrix electro-optical processor to perform

the DFT by Goodman, et al. [29] and for APAR processing by Psaltis, et al. [5]. In this technique, each element is decomposed into its residues along unit vectors spaced by $120^\circ = 2\pi/3$ in the complex plane as

$$\mathbf{x} = x_0 \exp(j\pi 0) + x_1 \exp(j2\pi/3) + x_2 \exp(j4\pi/3) \quad (4.16)$$

For simplicity, we denote the three components of \vec{x} along the three directions by \vec{x}_0 , \vec{x}_1 , and \vec{x}_2 , with similar notation used for the three components of the matrices \underline{H} and the output vectors \vec{v} .

In this notation, we describe Fig. 4.6 as

$$\begin{bmatrix} \vec{v}_0 \\ \vec{v}_1 \\ \vec{v}_2 \end{bmatrix} = \begin{bmatrix} H_0 & H_2 & H_1 \\ H_1 & H_0 & H_2 \\ H_2 & H_1 & H_0 \end{bmatrix} \begin{bmatrix} \vec{x}_0 \\ \vec{x}_1 \\ \vec{x}_2 \end{bmatrix} \quad (4.17)$$

In (4.17), all components are real and positive. The P_A input is $3M$ long, P_B is $3N \times 3M$ and P_C is $3N$. The P_2 mask must be properly arranged as shown in (4.17). The increased SBW of P_A - P_C by factors of 3 or 9 is the major disadvantage of such a system. This system arrangement was the one we used in our earlier phase 1 work [5] to realize and demonstrate electro-optical processing of complex-valued APAR data.

The second complex-valued data processing technique previously described [6] and demonstrated by us used color-multiplexing and the complex decomposition in (4.16). In this processor (see Fig. 4.7) three linear LD or LED input arrays are used at P_1 with different wavelength outputs λ_0 - λ_2 . The inputs to the three LD input arrays are \vec{x}_0 - \vec{x}_2 and the B mask at P_2 was arranged as

$$\underline{B} = \begin{bmatrix} H_0 & H_1 & H_2 \end{bmatrix} \quad (4.18)$$

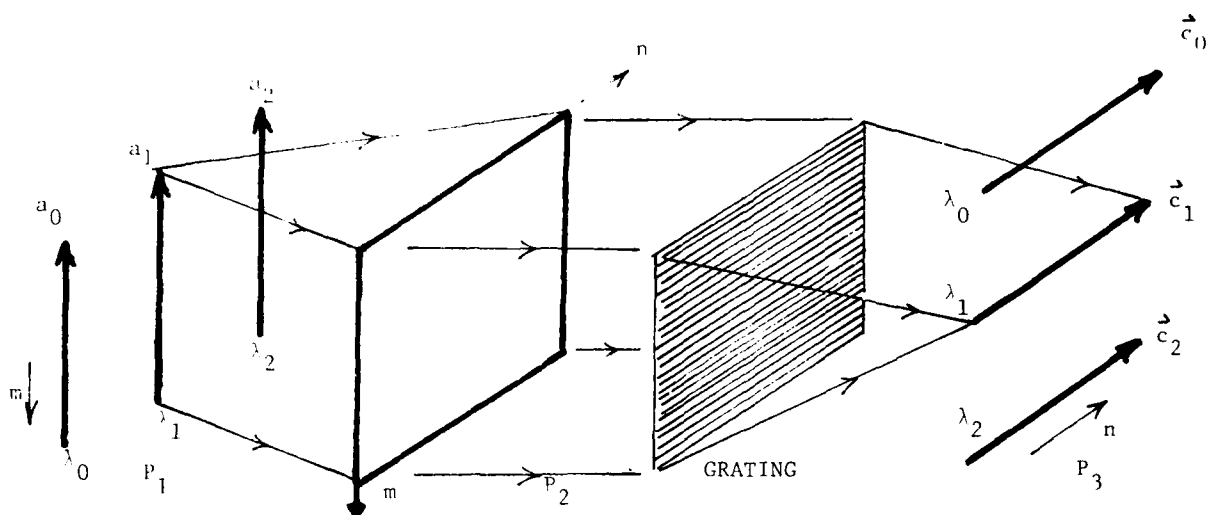


Fig. 4.7 Original color-multiplexed vector-matrix processor schematic.

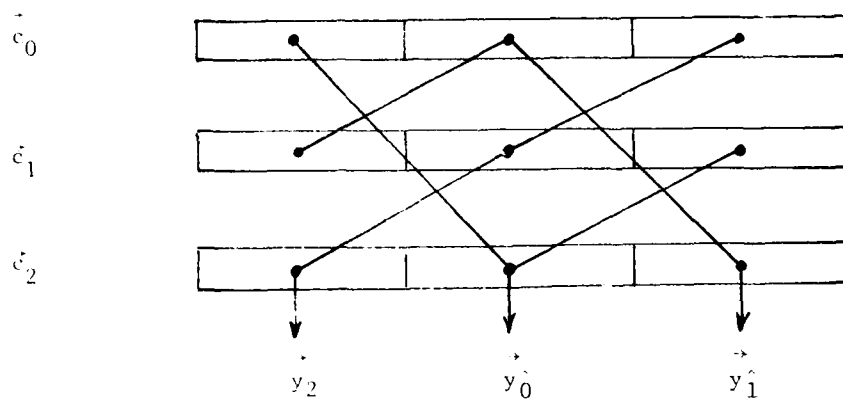


Fig. 4.8 Output detector post processing at P_3 of Fig. 2.7 to realize \vec{y}_0 , \vec{y}_1 , \vec{y}_2 from the \vec{c}_0 , \vec{c}_1 , \vec{c}_2 outputs.

The B mask was illuminated with all input wavelengths as before and the three separate AS were then separated vertically (not horizontally) at P_3 by a grating between P_2 and P_3 . The output P_3 patterns on the three linear output photodetector arrays are thus

$$\begin{aligned}\vec{c}_0 &= c_{0n} = [B_0, B_1, B_2]a_0 \\ \vec{c}_1 &= c_{1n} = [B_0, B_1, B_2]a_1 \\ \vec{c}_2 &= c_{2n} = [B_0, B_1, B_2]a_2\end{aligned}\quad (4.19)$$

and the system so described by

$$\begin{bmatrix} \vec{c}_0 \\ \vec{c}_1 \\ \vec{c}_2 \end{bmatrix} = \underline{B} [\vec{a}_0, \vec{a}_1, \vec{a}_2] \quad (4.20)$$

The desired $\vec{y}_0, \vec{y}_1, \vec{y}_2$ outputs are:

$$\begin{aligned}\vec{y}_0 &= H_0\vec{x}_0 + F_2\vec{x}_1 + H_1\vec{x}_2 \\ \vec{y}_1 &= H_1\vec{x}_0 + H_0\vec{x}_1 + H_2\vec{x}_2 \\ \vec{y}_2 &= H_2\vec{x}_0 + H_1\vec{x}_1 + H_0\vec{x}_2\end{aligned}\quad (4.21)$$

the observed $\vec{c}_0, \vec{c}_1, \vec{c}_2$ outputs at P_3 are:

$$\begin{aligned}\vec{c}_0 &= [B_0\vec{a}_0, B_1\vec{a}_0, B_2\vec{a}_0] \\ \vec{c}_1 &= [B_0\vec{a}_1, B_1\vec{a}_1, B_2\vec{a}_1] \\ \vec{c}_2 &= [B_0\vec{a}_2, B_1\vec{a}_2, B_2\vec{a}_2]\end{aligned}\quad (4.22)$$

To realize (4.21) from (4.22) we combine the detector array outputs as shown in Figure 4.8;

thus

$$\begin{aligned}\vec{y}_0 &= B_0 \vec{a}_0 + B_1 \vec{a}_2 + B_2 \vec{a}_1 \\ \vec{y}_1 &= B_0 \vec{a}_1 + B_1 \vec{a}_0 + B_2 \vec{a}_2 \\ \vec{y}_2 &= B_0 \vec{a}_2 + B_1 \vec{a}_1 + B_2 \vec{a}_0.\end{aligned}\tag{4.23}$$

The system of Fig. 4.7 reduces the 1-D input and output SBW requirements by a factor of three (by use of three 1-D input LD arrays at different λ and three 1-D output detector arrays stacked vertically) and the SBW of the P_2 mask B to $3 \cdot N \cdot M$. In Sect. 4.5, a superior color-multiplexed processor without electronic post processing is described.

4.5 COMPLEX-VALUED WAVELENGTH-DIVERSITY PROCESSOR

In this section, we describe new and preferable bipolar and complex-valued APAR processors and techniques using the wavelength-diversity processor. We also provide experimental confirmation and demonstration of the system.

We first consider the new bipolar-valued data processor to realize (4.15). The basic system of Fig. 4.1 is used with the data formatted as shown in Fig. 4.4. For this case, we decompose \vec{x} into \vec{x}^+ and \vec{x}^- and \underline{H} into \underline{H}^+ and \underline{H}^- . At P_A , we place three linear LD arrays each emitting at a different $\lambda_i = \lambda_0, \lambda_1$, and λ_2 . We describe the three linear LD outputs at P_A by $\vec{a}_0, \vec{a}_1, \vec{a}_2$ and set $\vec{a}_0 = \vec{x}^+, \vec{a}_1 = \vec{x}^-$, and $\vec{a}_2 = \vec{x}^+$. At P_B , we place two matrix masks side-by-side, $B_0 = \underline{H}^+$ and $B_1 = \underline{H}^-$. The output at P_C will thus have four terms spatially separated horizontally:

$$\begin{aligned}
 \vec{c}_0 &= B_0 \vec{a}_0 \\
 \vec{c}_1 &= B_1 \vec{a}_0 + B_0 \vec{a}_1 \\
 \vec{c}_2 &= B_1 \vec{a}_1 + B_0 \vec{a}_0 \\
 \vec{c}_3 &= B_1 \vec{a}_2.
 \end{aligned}
 \tag{4.24}$$

The central \vec{c}_1 and \vec{c}_2 outputs at P_C are the desired bipolar components of \vec{u} :

$$\begin{aligned}
 \vec{y}^- &= \vec{c}_1 = \underline{H} \vec{x}^+ + \underline{H}^+ \vec{x}^- \\
 \vec{y}^+ &= \vec{c}_2 = \underline{H} \vec{x}^- + \underline{H}^+ \vec{x}^+.
 \end{aligned}
 \tag{4.25}$$

From (4.25), we see that the use of one additional linear LD input array provides the correct \vec{y}^- and \vec{y}^+ bipolar output components on the two parts of a linear output photodetector array in P_C . Use of the prior color-multiplexing scheme with bipolar data would require electronic post-processing to sum and combine the proper photodetector outputs as well as two linear output detector arrays stacked vertically.

We next consider a second method of processing bipolar-valued data using wavelength diversity. This approach combines the methods of (4.14) and (4.15). This technique uses two linear LD input arrays (rather than three as above) and one P_B mask (not two as above). The LD outputs at λ_0 and λ_1 are $\vec{a}_0 = \vec{x}^+$ and $\vec{a}_1 = \vec{x}^-$ and the components of the B mask are $b_{m,n} = (b_{m,n} + 1/2)$. The resultant outputs on the two parts of a linear photodetector array at P_C are:

$$\begin{aligned}
 \vec{y}_n^+ &= \sum_m b_{m,n} a_{0,m} - \sum_m a_{0,m} / 2 \\
 \vec{y}_n^- &= \sum_m b_{m,n} a_{1,m} - \sum_m a_{1,m} / 2.
 \end{aligned}
 \tag{4.26}$$

We can generate the last two terms in (4.26) by use of an additional column of B with transmittance $1/2$ as noted earlier. The outputs from the two associated

photodetectors are the last terms in (4.26) and thus can be electronically subtracted from the observed c_m outputs to produce the desired \vec{y}^+ and \vec{y}^- outputs as in (4.26). Thus in order to reduce the SBW requirements of P_A and P_B , we trade off dynamic range in P_B (by adding a bias) and require some electronic post processing.

We now consider a wavelength diversity processor to realize the complex-valued vector-matrix operations required in APAR. This system does not require the electronic post-processing of our prior system [4.1] nor does it need 2-D or stacked 1-D output detector arrays.

We consider forming

$$\vec{y} = \underline{H}\vec{x} \quad (4.27)$$

where all elements can be complex. At P_A , we place five (not 3) linear LD arrays emitting at $\lambda_0 - \lambda_4$ and with inputs $\vec{a}_0 - \vec{a}_4$, where

$$\vec{a}_0 = \vec{x}_0, \vec{a}_1 = \vec{x}_1, \vec{a}_2 = \vec{x}_2, \vec{a}_3 = \vec{x}_0, \vec{a}_4 = \vec{x}_1, \quad (4.28)$$

and $\vec{x}_0 - \vec{x}_2$ are described by (4.16). At P_B , three \underline{B} matrices $B_0 - B_2$ are placed side by side where

$$\underline{B}_0 = \underline{H}_0, \underline{B}_1 = \underline{H}_1, \underline{B}_2 = \underline{H}_2, \quad (4.29)$$

and $\underline{H}_0 - \underline{H}_2$ are the projections of \underline{H} as in (4.16). The resultant P_C pattern contains terms in five wavelengths.

At λ_0 , we find $\underline{B}\vec{a}_0$ containing the three terms $\underline{B}_0\vec{a}_0$, $\underline{B}_1\vec{a}_0$ and $\underline{B}_2\vec{a}_0$ separated by Δp_c , which was chosen equal to the Δp_b separation of \underline{H}_0 , \underline{H}_1 and \underline{H}_2 at P_B . The $\underline{B}\vec{a}_1$ output at λ_1 is shifted right by Δp_c from the $\underline{B}\vec{a}_0$ output with the $\underline{B}\vec{a}_2$ output shifted right by Δp_c from the $\underline{B}\vec{a}_1$ output. The $\underline{B}\vec{a}_3$ output equals $\underline{B}\vec{a}_0$ and is displaced from the $\underline{B}\vec{a}_2$ output by Δp_c (all $\lambda_{l+1} - \lambda_l$ values are assumed to equal $\Delta\lambda$ and $\Delta\lambda$ is assumed to correspond to a Δp_c). We show the $\underline{B}\vec{a}_3 = \underline{B}\vec{a}_0$ output on the top line in the P_C pattern in Fig. 4.9, displaced by Δp_c from the $\underline{B}\vec{a}_2$ pattern and by $3\Delta p_c$ from the $\underline{B}\vec{a}_0$ pattern in λ_0 light. The $\underline{B}\vec{a}_4$ pattern is similarly shown beside the $\underline{B}\vec{a}_1$ pattern since $\underline{B}\vec{a}_4 = \underline{B}\vec{a}_1$. It is displaced by $3\Delta p_c$ from the λ_1 pattern for reasons noted earlier.

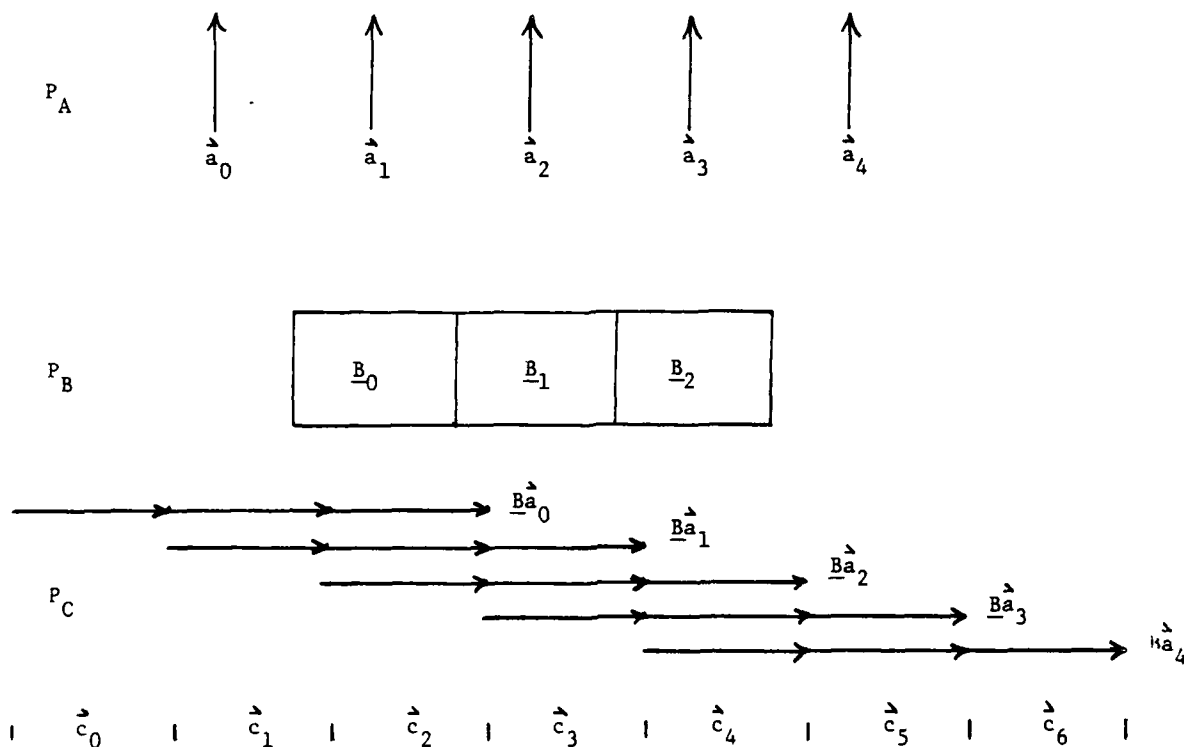


Fig. 4.9 Formats for the novel complex-valued data processor for APAR.

As before, the actual patterns at P_C are superimposed vertically and are shown separated vertically only for clarity. The P_C detector forms the sum vertically in m of the P_C outputs shown. The outputs of the three central detector arrays are

$$\begin{aligned}\hat{c}_2 = \hat{y}_2 &= H_2 \hat{x}_0 + H_1 \hat{x}_1 + H_0 \hat{x}_2 \\ \hat{c}_3 = \hat{y}_0 &= H_0 \hat{x}_0 + H_2 \hat{x}_1 + H_1 \hat{x}_2 \\ \hat{c}_4 = \hat{y}_1 &= H_1 \hat{x}_0 + H_0 \hat{x}_1 + H_2 \hat{x}_2.\end{aligned}\quad (4.30)$$

We note that each of the output components (\hat{y}_0 , \hat{y}_1 , and \hat{y}_2) in (4.30) contain three P_C . From (4.30) we see that the outputs from the $L \cdot N = 3N$ linear output detector elements are the $\hat{0}$, $\hat{1}$ and $\hat{2}$ projections of the complex \hat{y} vector-matrix product in (4.27). Thus, by the use of five input LD arrays (rather than 3), no output P_C electronic processing (summing and reordering of data) is needed to obtain the components of the complex output (as in Fig. 4.8).

To demonstrate the use of the data formats in Fig. 4.9 in the system of Fig. 4.1, a specific complex vector x and matrix H were chosen:

$$\hat{x} = \begin{bmatrix} 1 + \exp(j2\pi/3) \\ 1 + \exp(j4\pi/3) \\ 1 \\ \exp(j2\pi/3) \\ \exp(j4\pi/3) \end{bmatrix} = \begin{bmatrix} \hat{0} + \hat{1} \\ \hat{0} + \hat{2} \\ \hat{0} \\ \hat{1} \\ \hat{2} \end{bmatrix} \quad (4.31)$$

$$\underline{H} = \begin{bmatrix} \hat{0} & \hat{1} & \hat{0} + \hat{2} & \hat{1} + \hat{2} & \hat{2} \\ \hat{1} & \hat{0} + \hat{2} & \hat{1} + \hat{2} & \hat{2} & \hat{0} \\ \hat{0} + \hat{2} & \hat{1} + \hat{2} & \hat{2} & \hat{0} & \hat{1} \\ \hat{1} + \hat{2} & \hat{2} & \hat{0} & \hat{1} & \hat{0} + \hat{2} \\ \hat{2} & \hat{0} & \hat{1} & \hat{0} + \hat{2} & \hat{1} + \hat{2} \end{bmatrix} \quad (4.32)$$

The short-hand $\hat{0}$, $\hat{1}$, $\hat{2}$ notation is used for simplicity of description. We write the three $\hat{0}$, $\hat{1}$, $\hat{2}$ components of \vec{x} and \underline{H} explicitly as

$$\vec{x}_0 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \quad \vec{x}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \quad \vec{x}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (4.33)$$

$$\underline{H}_0 = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix}, \quad \underline{H}_1 = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix}, \quad \underline{H}_2 = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 \end{bmatrix} \quad (4.34)$$

The format of Fig. 4.9 was thus used in the laboratory version in Fig. 4.5 of the system of Fig. 4.1 with the vector-matrix data of (4.33) and (4.34). The P_A pattern is shown in Fig. 4.10a for the vector in (4.33). Each of the five vector inputs shown are actually in different colors or λ_k but are printed in black and white. The P_B pattern containing \underline{H}_0 , \underline{H}_1 and \underline{H}_2 in (4.34) is shown in Fig. 4.10b. The output P_C pattern is shown in parts in Fig. 4.10c. The output in colors λ_0 and λ_3 is shown first. Each is $\underline{H} \vec{x}_0$. Because $\lambda_3 - \lambda_0 = 3\Delta\lambda$ corresponds to a $3p_c$ separation, the λ_0 and λ_3 pattern contains two

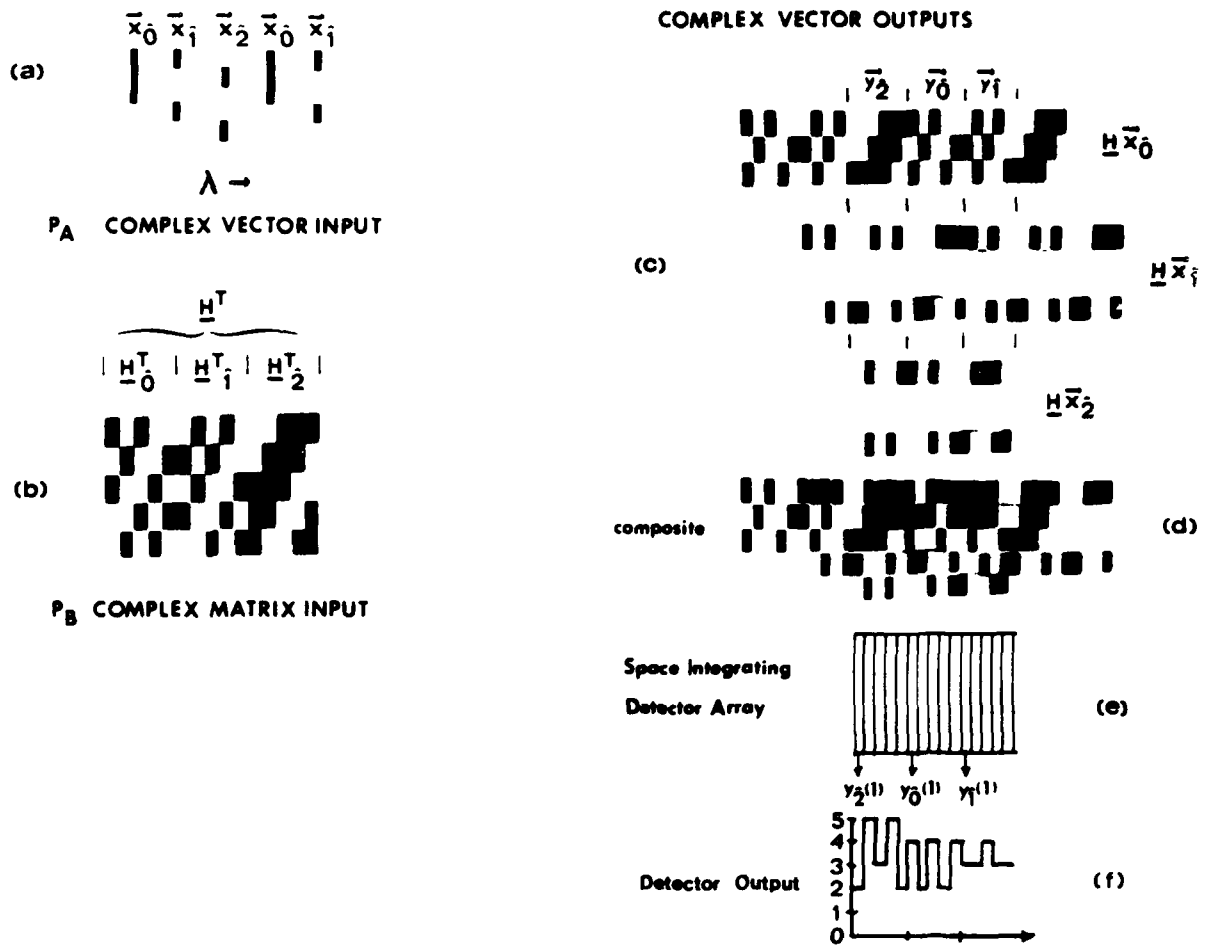


Fig. 4.10 Example of complex-valued vector-matrix multiplication.

$\underline{H} \vec{x}_0$ patterns side-by-side. Since \vec{x}_0 contains three 1's, only the first three rows of \underline{H} appear as shown.

The output pattern in λ_1 and λ_4 is shown next. Since $\vec{a}_1 = \vec{a}_4 = \vec{x}_1$, this pattern is two $\underline{H} \vec{x}_1$ patterns side by side. Since \vec{x}_1 contains a 1 as the first and fourth elements, only the first and fourth rows of \underline{H} appear in the $\underline{H} \vec{x}_1$ output as shown. The $\underline{H} \vec{x}_2$ output is shown next.

As noted before, these patterns are superimposed vertically at P_C . This composite P_C output pattern is shown in Fig. 4.10d. The central 15 columns of the Fig. 4.10d pattern are incident upon a 15 element photodetector array shown in Fig. 4.10e. The detector integrates each column in Fig. 4.10d. The first 5 detector outputs in these 15 elements are the five elements of \vec{y}_2 , the central 5 detector outputs are the elements of \vec{y}_0 and the right 5 detector outputs are the elements of the \vec{y}_1 component of the complex $\vec{y} = \underline{H} \vec{x}$ output.

When the system of Fig. 4.1 is used as an iterative processor, a 5 element complex vector with components $\hat{0}$, $\hat{1}$, $\hat{2}$ is added to the respective 15 detector outputs and fed back to P_A as the $\hat{0}$, $\hat{1}$, $\hat{2}$ components.

To show that the experimentally obtained output is correct for the specific example in (4.33) and (4.34), we computed $\vec{y} = \underline{H} \vec{x}$ and found

$$\begin{aligned} \vec{y} &= [y_0, y_1, y_2, y_3, y_4] \\ &= [4\hat{0} + 3\hat{1} + 2\hat{2}, 2\hat{0} + 3\hat{1} + 5\hat{2}, 4\hat{0} + 4\hat{1} + 3\hat{2}, 2\hat{0} + 3\hat{1} + 5\hat{2}, 4\hat{0} + 3\hat{1} + 2\hat{2}] \quad (4.35) \\ &= [2\hat{0} + \hat{1}, \hat{1} + 3\hat{2}, \hat{0} + \hat{1}, \hat{1} + 3\hat{2}, 2\hat{0} + \hat{1}]. \end{aligned}$$

The first representation in (4.35) lists the five elements of \vec{y} . The computation of $\vec{y} = \underline{H} \vec{x}$ yields the second expression in (4.35). Since $\hat{0} + \hat{1} + \hat{2} = 0$, we subtract these common factors and obtain the last expression in (4.35).

The detector outputs from left to right are

$$\vec{y} = \vec{y}_2, \vec{y}_0, \vec{y}_1 = y_{20} \cdots y_{24} \ y_{00} \cdots y_{04} \ y_{10} \cdots y_{14}. \quad (4.36)$$

From (4.35), the expected P_C detector output is

$$\bar{y} = \bar{y}_2 \ \bar{y}_0 \ \bar{y}_1 = [030302010211111]. \quad (4.37)$$

The optical output data (obtained by summing the column outputs in Fig. 4.10d) is

$$\begin{array}{c|c|c} 25352 & 42424 & 33433 \end{array} = \vec{y}_2, \vec{y}_0, \vec{y}_1. \quad (4.36a)$$

Combining the first elements of \vec{y}_2 , \vec{y}_0 and \vec{y}_1 and then the second etc elements, we obtain

$$243 \quad 523 \quad 344 \quad 523 \quad 243.$$

The symmetry of these outputs was set by the choice of matrix used.

Subtracting the offset or lowest value from each pair of three values (e.g. for the first set 232, the lowest value is 2. Subtracting 2 from each element of this set yields 012) yields

$$021 \quad 301 \quad 011 \quad 301 \quad 021. \quad (4.36b)$$

We now recall that the three elements in each of the above groupings correspond to the 2 0 1 component values respectively, we write the above output as

$$2 \ 0 + 1 \quad 3 \ 2 + 1 \quad 0 + 1 \quad 3 \ 2 + 1 \quad 2 \ 0 + 1. \quad (4.37)$$

This detector output pattern is shown in Fig. 4.10f after electronic post processing. This pattern in (4.37) agrees with the predicted result in (4.35), thus verifying the system's operation as a vector-matrix multiplier of complex-valued data for APAR. The subtraction of the smallest component of each y_n element output from all five of those output components is easily achieved by a comparator/subtractor. This is necessary because the components 0, 1, 2 are not linearly independent. A simple geometrical construction to allow computation of the components of the complex data is shown in Fig. 4.11.

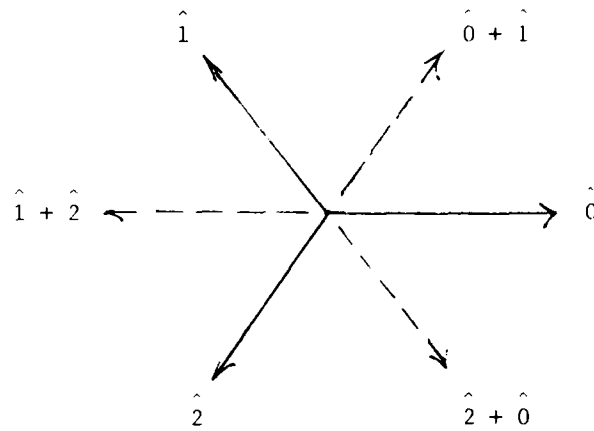


Fig. 4.11 Geometrical construction to determine the product of complex vectors as projection components $\hat{0}$, $\hat{1}$, $\hat{2}$.

4.6 MULTICHANNEL 1-D CONVOLVER

An alternate and more general formulation of the wavelength diversity processor of Fig. 4.1 as a multi-channel 1-D or vector-matrix convolver is possible. We can advance this formulation with reference to Fig. 4.4. We denote the L linear vectors at P_A by the index ℓ , the R matrices at P_B by the index r , and the L linear N element outputs at P_C by the index i also. Summing the vector outputs at $\ell = 2$ in P_C , we obtain

$$\vec{c}_i = \vec{c}_2 = \underline{B}_2 \vec{a}_0 + \underline{B}_1 \vec{a}_1 + \underline{B}_0 \vec{a}_2 \quad (4.38)$$

or in more general notation with ℓ and r as the \vec{a} and \underline{B} subscripts,

$$\vec{c}_\ell = \sum_r \underline{B}_r \vec{a}_{\ell-r} \quad (4.39)$$

For $\ell = 2$ and $r = 0, 1, 2$, (4.39) becomes

$$\vec{c}_2 = \sum_{r=0}^2 \underline{B}_r \vec{a}_{2-r} = \underline{B}_0 \vec{a}_2 + \underline{B}_1 \vec{a}_1 + \underline{B}_2 \vec{a}_0. \quad (4.40)$$

By inspection of (4.39) and Fig. 4.4, we see that the form of the output at a given N element detector in P_C is that of a type of convolver, specifically a multi-channel 1-D vector-matrix convolution. This general operational description in (4.39) of the system improves the power and flexibility of the wavelength-diversity processor with wavelength λ as the convolution shift variable. We thus rewrite (4.39) as

$$\begin{aligned} c(\lambda) &= \int b(r) a(\lambda-r) dr \\ c(\lambda) &= \int o(r) a(\lambda-r) dr, \end{aligned} \quad (4.41)$$

from which the convolution with λ as the shift variable is apparent. The P_C output is shown in Fig. 4.12 for this general case.

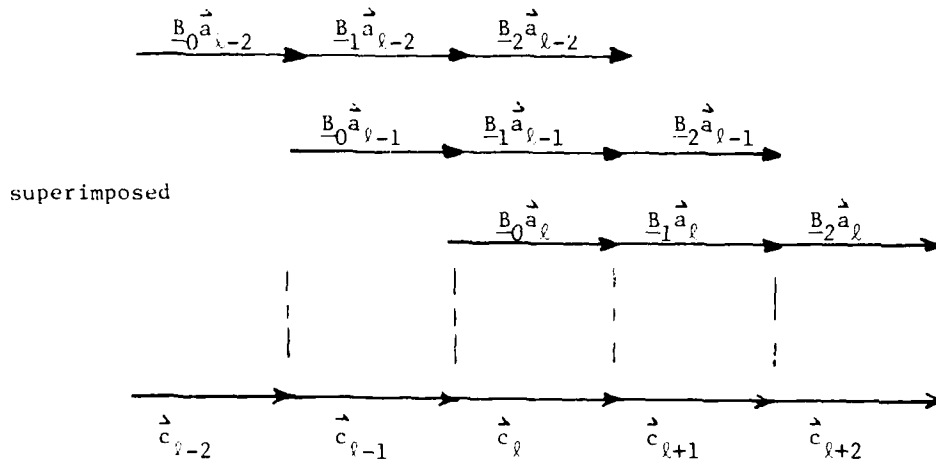


Fig. 4.12 General P_C pattern description as a multi-channel 1-D wavelength convolver.

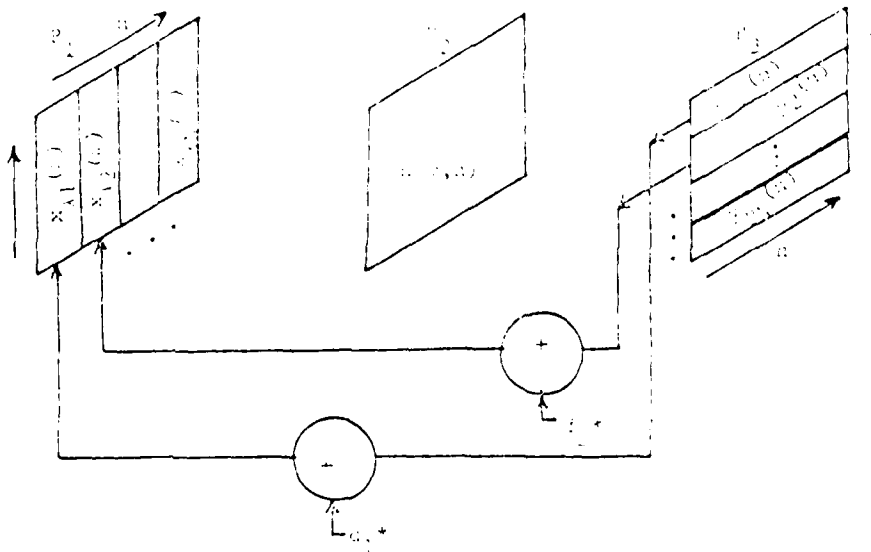


Fig. 4.13 Example of matrix-matrix multiplication.

4.7 MATRIX-MATRIX MULTIPLICATION

In this section, we consider a new application of the wavelength diversity processor, to perform a matrix-matrix multiplication. We describe two new electro-optical systems that achieve this operation. Consider formation of

$$\underline{C} = \underline{B} \underline{A}, \quad (4.42)$$

where

$$\underline{C} = \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix}, \quad \underline{B} = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}, \quad \underline{A} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}. \quad (4.43)$$

Substitution shows

$$\begin{aligned} \underline{B} \underline{A} &= \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \\ &= \begin{pmatrix} b_{11} a_{11} + b_{12} a_{21} & b_{11} a_{12} + b_{12} a_{22} \\ b_{21} a_{11} + b_{22} a_{21} & b_{21} a_{12} + b_{22} a_{22} \end{pmatrix} \\ &= \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix} = \underline{C} \end{aligned} \quad (4.44)$$

to be the desired result.

Consider the simplified version of Fig. 4.1 shown in Fig. 4.13. The input at P_1 is N vectors $a_{n(m)}$ each of length M corresponding to N spatially separated wavelengths λ_n . Each linear LD vector is imaged vertically and spread horizontally across P_2 , where a mask $B(m,n)$ is placed. P_2 is imaged onto P_3 , while each λ_n is separated vertically

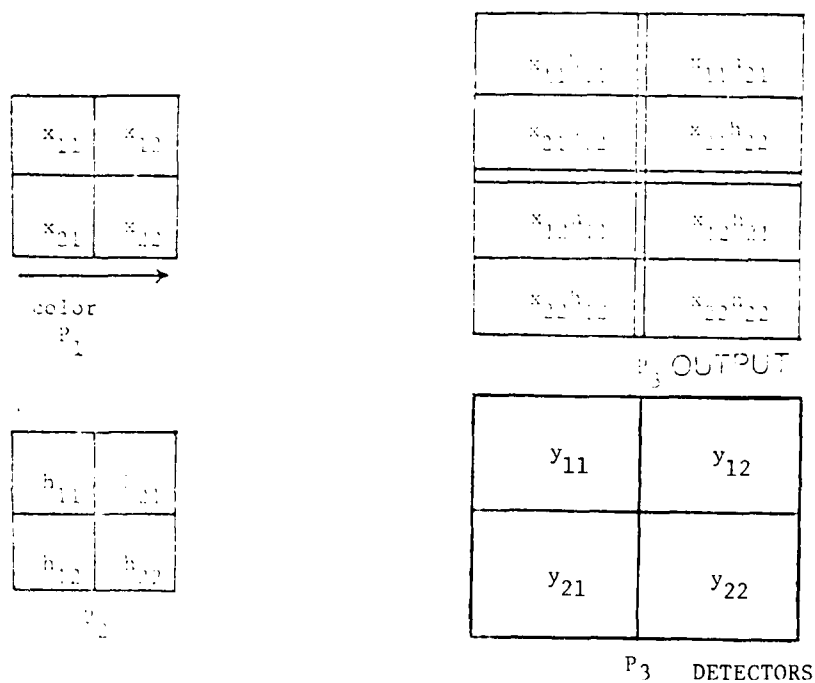


Fig. 4.14 Formats for the data planes in Fig. 4.13 for matrix-matrix multiplication.

at P_3 by a grating as before. The P_3 pattern is thus described by N vectors $c_{pn}(n)$ each of length N and each formed in a different λ_n light where $p = \lambda f_s z$, (λ is the input wavelength, f_s is the spatial frequency of the grating between P_2 and P_3 and z is the distance from P_2 to P_3).

To realize (4.44) using Fig. 4.13, we place A at P_1 as shown in Fig. 4.14. When the B^T mask is placed at P_2 , the P_3 output containing eight elements results. The detector array at P_3 contains four elements as shown. Overlaying the detector and P_3 output patterns, we see that each detector integrates the sum of the corresponding two output P_3 components. The four detector outputs are thus the components of the desired C output matrix in transposed form.

We now consider a matrix-matrix multiplier using a 1-D detector and 1-D input vectors of longer length. To visualize this system, recall the description of Fig. 4.1 in (4.1),

$$c_{pn} = \sum_m^M b_{mn} a_{ma} \quad (4.1)$$

This is the product of the two matrices A and B ,

$$C = BA, \quad (4.45)$$

where A is an $L \times M$ matrix and B is $N \times M$.

To demonstrate this use of the system of Fig. 4.1, we consider the matrices

$$A = \begin{matrix} & \xrightarrow{m} \\ \downarrow \\ \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{bmatrix} \end{matrix} \quad B = \begin{matrix} & \xrightarrow{n} \\ \downarrow \\ \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{bmatrix} \end{matrix} \quad (4.46)$$

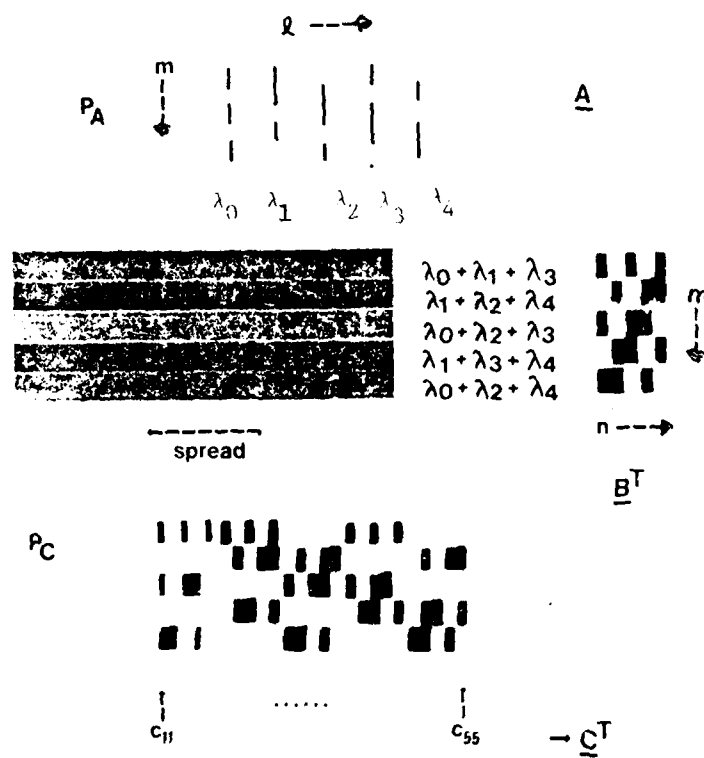


Fig. 4.15 Example of new matrix-matrix multiplication algorithm.

Substitution into (4.45) yields

$$C = \begin{bmatrix} 3 & 1 & 2 & 2 & 1 \\ 1 & 2 & 2 & 1 & 3 \\ 2 & 2 & 1 & 3 & 1 \\ 2 & 1 & 3 & 1 & 2 \\ 1 & 3 & 1 & 2 & 2 \end{bmatrix} . \quad (4.47)$$

To perform the matrix-matrix product in (4.46) and (4.47) on the system of Fig. 4.1, the experimental system of Fig. 4.5 was used with five different input wavelengths to distinguish the five rows of A. The input P_A pattern is shown in the top of Fig. 4.15, in actuality this image was in color with each column being a different color. This 2-D input pattern with color horizontal with index n and with five elements per column with index m describes the 2-D matrix $A(n,m)$. Incident on P_B , the five rows are illuminated with different combinations of input wavelengths as shown in the center of Fig. 4.15. The P_B mask $B(n,m)$ in (4.46) is shown to the center right in Fig. 4.15. The P_2 output plane pattern at P_C is shown in the bottom of Fig. 4.15.

The detector at P_C sums the five entries in each column in P_C . The 25 detector outputs (from inspection of the bottom figure in Fig. 4.15) are (from left-to-right)

$$c_{11} \cdots c_{55} = 3122112213221312131213122. \quad (4.48)$$

We see that these detector outputs are the elements of the matrix C in (4.47) thus proving the system's use as a matrix-matrix multiplier.

4.8 MATRIX-INVERSION

We now consider yet another new application of the wavelength-diversity processor, its use in inverting a matrix. This application is of direct use in APAR processing. Consider the matrix-matrix multiplier of Fig. 4.13 as described in Sect. 4.7 with the feedback shown applied. We describe this IOP matrix processor with P_A input $\underline{W}_i - 1$, P_B mask $[H]$, output \underline{W}_i . After addition of a matrix \underline{F} to the P_C output, the IOP system is described by

$$\underline{W}_i + 1 = \underline{H} \underline{W}_i + \underline{F}, \quad (4.49)$$

where all quantities are matrices and the subscripts refer to successive output iterations. In steady state $\underline{W}_i = \underline{W}_i + 1 = \underline{W}$ and (4.49) becomes

$$\begin{aligned} \underline{W} &= \underline{H} \underline{W} + \underline{F} \\ &= \underline{F}(\underline{I} - \underline{H})^{-1} \end{aligned} \quad (4.50)$$

If we let $\underline{H} = \underline{I} - \underline{M}$ and $\underline{F} = \underline{I}$, then the output is

$$\underline{W} = \underline{M}^{-1}. \quad (4.51)$$

The system has thus performed the inversion of the matrix \underline{M} at P_B . The system and algorithm used are shown in the functional diagram in Fig. 4.16. Note that the output matrix to be added is the identity matrix and thus this potentially complex operation is greatly simplified.

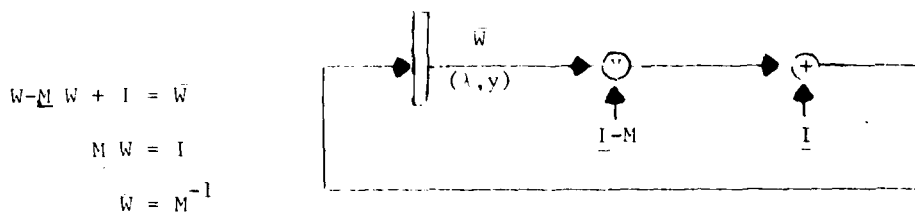


Fig. 4.16 Functional diagram of the matrix-inversion processor.

4.9 COVARIANCE MATRIX COMPUTATION

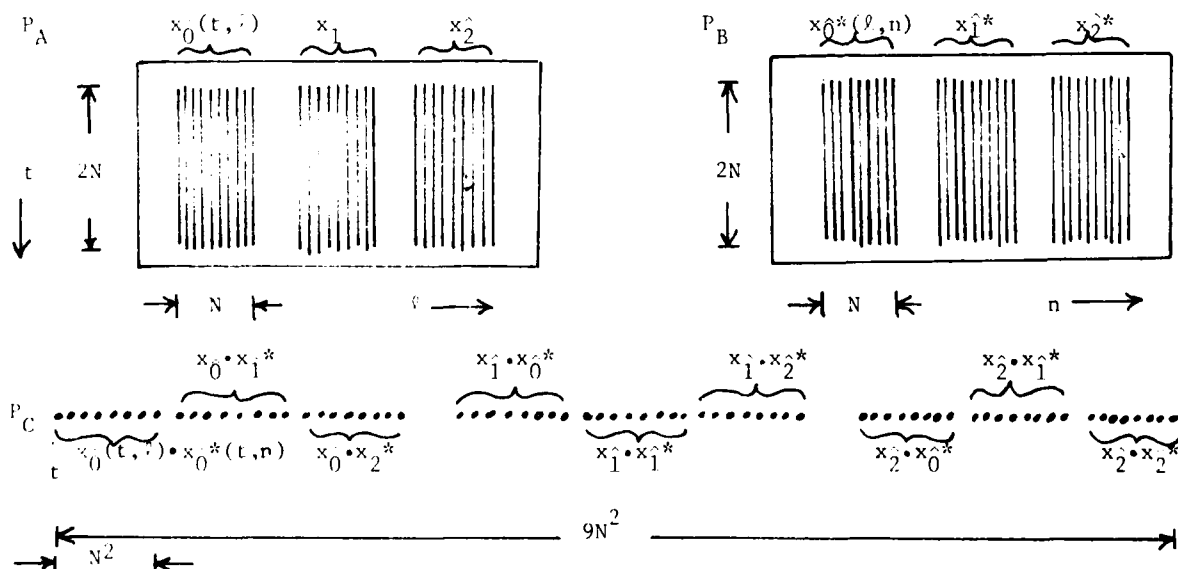
A difficult processing step in the APAR system involves the calculation of the covariance matrix, $M(\ell, n)$ of the noise field, namely:

$$M(\ell, n) = x_\ell(t)x_n^*(t) = \frac{1}{T} \int_T x_\ell(t)x_n^*(t)df \quad (4.52)$$

where $x_\ell(t)$ and $x_n(t)$ are the noise voltages received by the ℓ th and n th antenna elements. Rewriting (4.52) as a summation over t we obtain:

$$M(\ell, n) = \frac{1}{T} \sum_t x(t, \ell)x^*(t, n). \quad (4.53)$$

We note that (4.53) is identical in form to (4.1) and therefore conclude that in principle, the matrix-matrix multiplier can be used to compute the covariance matrix by writing the time history of each of the $\ell, n = 0, 1, 2 \dots N-1$ received signals along the m spatial coordinate in P_A and P_B where $x_\ell(t)$ is written along the ℓ th column in P_A and $x_n^*(t)$ is written along the n th column in P_B . Since the received signals are complex valued, the SBWP required in P_A and P_B to represent the data must be 3X as large and the resultant output P_C SBWP must be 9X larger than that required to perform a matrix-matrix multiplication of positive valued data. The P_A , P_B and P_C data formats would be:



where N is the number of antenna elements and $2N$ is the minimum number of time samples required to estimate the covariance matrix \underline{M} as a time average.

Since the above calculation is essentially a matrix-matrix multiplication, an experimental demonstration was not conducted. Further experimental work is recommended in this area in order to assess the accuracy of the overall APAR processing system.

4.10 SUMMARY

In this chapter, we have described a new wavelength-diversity processor. The use of this system as a vector-matrix multiplier and iterative processor with complex-valued and bipolar valued data elements was presented. Three new optical systems using three new algorithms to represent and process complex and bipolar data in non-coherent electro-optic vector-matrix processors were described.

The system's use as a matrix-matrix multiplier, a matrix inverter, a 1-D multichannel convolver and for computation of the covariance matrix M were also described. In each case, new system data arrangements were used to realize new and more powerful electro-optical processors and new adaptive algorithms for various APAR processing applications.

A laboratory wavelength-diversity electro-optical processor was assembled. It was used to experimentally demonstrate and verify its use in complex-valued vector-matrix multiplication and as a matrix-matrix multiplier. In all instances, experiments and theory agreed. The system offers the promise of increased processor capacity (due to the added degree of freedom that wavelength provides) and is of use in developing new algorithms (as the systems described, fabricated and demonstrated have shown).

The highlights of the WDP approach to APAR and the new research performed during the past year include:

- (1) WDP concept and demonstration of it.
- (2) Use of the WDP concept in a new complex-valued data representation (Sect. 4.4).

- (3) Modification of the original WDP concept to decrease the electronic post-processing necessary and demonstration of this concept (Sect. 4.5).
- (4) Laboratory WDP system formulation and demonstration using an arc lamp and grating (Sect. 4.3).
- (5) Multi-channel convolver description and demonstration of the WDP system (Sect. 4.6).
- (6) Matrix-matrix multiplication description and demonstration of the WDP system (Sect. 4.7).
- (7) Description of the use of the WDP system for matrix inversion (Sect. 4.8).
- (8) Description of the use of the WDP for covariance matrix computation (Sect. 4.9).

CHAPTER 5 SUMMARY AND CONCLUSION

We summarize the highlights of our programs in the area of coherent optical correlator (COC), iterative optical processor (IOP) and wavelength diversity processor (WDP) and their use in APAR processing.

Our new COC work on APAR processing has resulted in many new algorithms and system architectures as well as in the fabrication of many components for the COC processor. In the past one year we have:

- (1) Developed a new COC concept using 1-D AO cells rather than 2-D SLMs (because they are more easily fabricated and more readily available) and TI rather than SI correlators (because their longer integration times allow better noise statistical estimates) (Sect. 2.2).
- (2) Fabricated and performed initial testing of two AO cells (Sect. 2.3).
- (3) Developed a new simulator to handle multiple phased array signals with adaptivity in space and frequency, to compute weights and calculate corrected antenna patterns, plus conventional Fourier transform and correlation routines (Sect. 2.5).
- (4) Devised a new adjunct antenna concept that allows full space and frequency APAR data to be obtained using only a two-channel processor (Sect. 2.4).

- (5) Developed a new hybrid time and space integrating AO correlator architecture that combines the best features of the TI and the SI systems, and provides a 2-D display of the angle and frequency location of the far-field antenna noise pattern from the two-element adjunct antenna data (Sect. 2.6).
- (6) Designed and fabricated a hardware electronic support system and designed and nearly completed fabrication of a multi-purpose computer-driven electronic support system for complex noise source scenarios (Sect. 2.7).
- (7) Developed a self-correcting post-processing system, a new fast iterative modified projection method technique and an advanced projection concept to compute the adaptive weights for narrow-band and wide-band jammers that differ in both angle and frequency (Sect. 2.8).
- (8) Performed an experimental verification of a TI correlator and used it to demonstrate residue arithmetic computations in a new time position coding scheme. Residue arithmetic is an advanced technique whereby numerical computations can be performed in parallel with no carries and with reduced dynamic range requirements. This is necessary to realize an accurate optical computer (Sect. 2.9).
- (9) Demonstrated the use of the TSI system for computation of the 2-D space and frequency output antenna pattern from an adaptive array using an adjunct antenna (Sect. 2.10).

Thus, in conclusion, we have devised a new COC optical processor, demonstrated and simulated the key points of the system, begun development of the necessary post-processor, and fabricated the necessary components and support system to fully study this COC technique for APAR.

The IOP concept for APAR processing has proven to be most excellent. The highlights of our recent work on this approach to APAR processing are summarized below.

- (1) Design and fabrication of a new IOP system with fiber optic interconnections, pulse width modulation and a microprocessor electronic feedback system (Sect. 3.3).
- (2) Use of a new technique to allow the IOP to operate on complex-valued data (Sect. 3.2.4).
- (3) Development and demonstration of a new IOP simulator, antenna model, and 2-D to 1-D mapping with space and time adaptivity and including IOP error sources (Sect. 3.4).
- (4) Demonstration and measurement of the excellent accuracy of the IOP system to be less than 0.8% and with less than 1% error in the antenna SNR obtained (Sect. 3.5.1).
- (5) Development and demonstration of a new technique to ensure convergence to the steady state solution and fast convergence of the IOP system (Sect. 3.5.2).
- (6) Real-time demonstration and analysis of the laboratory IOP system for antenna adaptivity in space and time (Sect. 3.6).

The weights and antenna SNRs obtained experimentally were within 1% of the theoretical limit. This excellent system performance in the IOP laboratory system fabricated represents a major new optical data processing architecture that appears to be quite attractive and useful for APAR processing and other applications.

The highlights of the WDP approach to APAR and the new research performed during the past year include:

- (1) WDP concept and demonstration of it.
- (2) Use of the WDP concept in a new complex-valued data representation (Sect. 4.4).
- (3) Modification of the original WDP concept to decrease the electronic post-processing necessary and demonstration of this concept (Sect. 4.5).
- (4) Laboratory WDP system formulation and demonstration using an arc lamp and grating (Sect. 4.3).
- (5) Multi-channel convolver description and demonstration of the WDP system (Sect. 4.6).
- (6) Matrix-matrix multiplication description and demonstration of the WDP system (Sect. 4.7).
- (7) Description of the use of the WDP system for matrix inversion (Sect. 4.8).
- (8) Description of the use of the WDP for covariance matrix computation (Sect. 4.9).

REFERENCES

1. IEEE-F, 127 (August 1980) Special Issue on Phased Arrays.
2. APAR Report, D. Casasent, 1979.
3. D. Casasent, et al., SPIE (October 1979).
4. D. Casasent, EASCON (October 1980).
5. D. Psaltis, et al., SPIE, 180 (1979).
6. D. Psaltis, et al., Optics Letters (November 1979).
7. R. Sprague, et al., Applied Optics, 15, 89 (1976).
8. I. Chang, IEEE, SU-23, 2 (1976).
9. SPIE, Volume 214 (1979) Special Issue on Bulk Acousto Optics.
10. A. Papoulis, Probability, Random Variables and Stochastic Processes, McGraw-Hill, New York (1965).
11. L. Rabiner and B. Gold, Theory and Application of Digital Signal Processing, Prentice-Hall, New Jersey (1975).
12. B. Kumar and D. Casasent, Applied Optics, (Accepted).
13. L. Reed, et al., IEEE, AES-10, 853 (1974).
14. D. Boroson, IEEE, AES-16, 446 (1980).
15. H. Cramer, Mathematical Methods of Statistics, Princeton University Press (1946).
16. S. Prasad, IEEE, AP-28, 328 (1980).
17. R. Ramakrishnam, et al., Applied Optics, 18, 464 (1979).
18. J. von Neuman, The Geometry of Orthogonal Spaces, Volume 2, Princeton New Jersey, Princeton University Press (1950).
19. F. Takao and K. Komiyama, IEEE, AES-16, 452 (1980).
20. G. Frost, IEEE Proc., 60, 926 (1972).
21. A. Kuang, et al., Applied Optics, 18, 148 (1979).
22. D. Psaltis, et al., Applied Optics, 18, 163 (1979).
23. I. Crow, et al., Applied Optics, 17, (February 1978).

AD-A105 129

CARNEGIE-MELLON UNIV PITTSBURGH PA DEPT OF ELECTRICAL--ETC F/6 17/9
OPTICAL PROCESSING FOR ADAPTIVE PHASED ARRAY RADAR.(U)
JUN 81 D CASASANT

F30602-80-C-0048

UNCLASSIFIED

RADC-TR-81-152

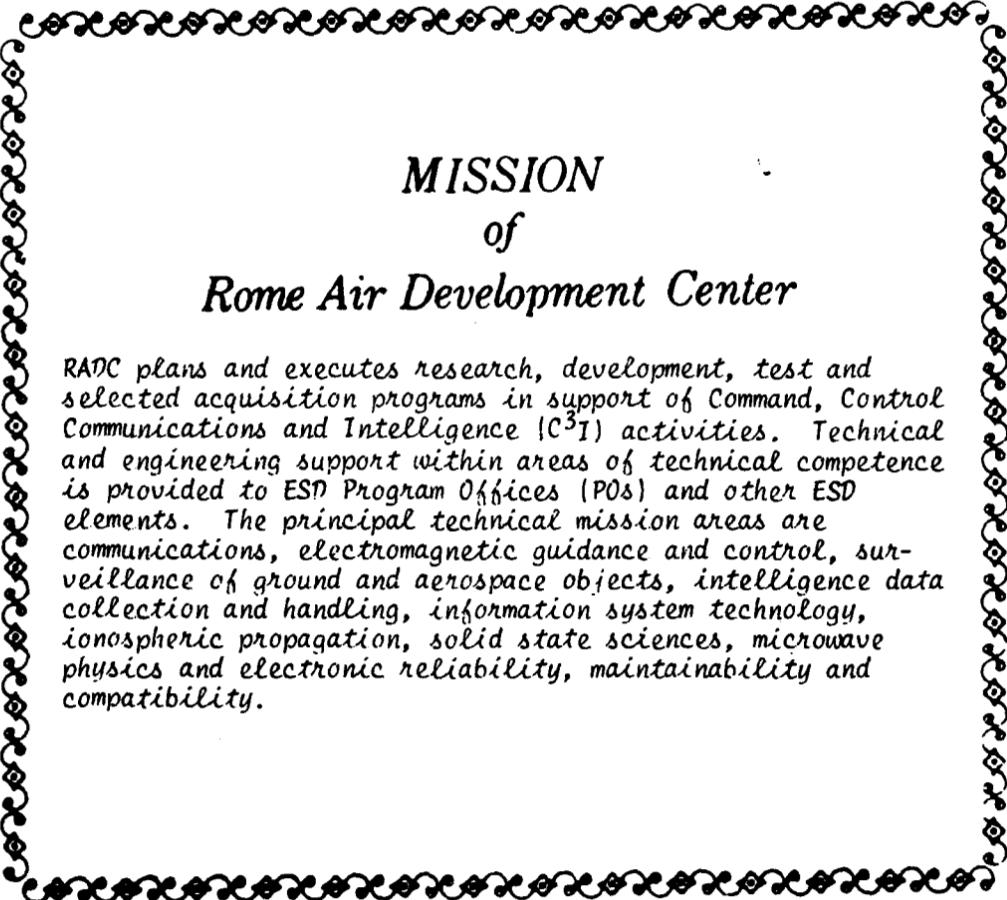
NL

3 of 3
AD
ADP/C



END
DATE
FILMED
10-81
DTIC

24. H. Kressel and J. Butler, Semiconductor Laser and Heterojunction LEDs, Academic Press.
25. S. Horiuchi, et al., IEEE Proc., (February 1978).
26. Bell Laboratories (Private Communications).
27. J. Bond, "Structural Organization for Real and Complex Data Convolution by Imaging CCDs", ARPA Quarterly Report Four, (1974).
28. J. W. Goodman, et al., Applied Optics, 16 (October 1977).
29. A. Warner, et al., J. Appl. Phys., 43, 4489 (1972).



*MISSION
of
Rome Air Development Center*

RADC plans and executes research, development, test and selected acquisition programs in support of Command, Control Communications and Intelligence (C³I) activities. Technical and engineering support within areas of technical competence is provided to ESD Program Offices (POs) and other ESD elements. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.